Exercise Set 1

Problem 1

Consider the following linear regression model:

$$y_i = \beta_1 + \beta_2 x_i + \varepsilon_i,$$

where $\beta = (\beta_1, \beta_2)'$ is a vector of unknown parameters, and x_i is a one-dimensional observable variable. We have a sample of $i = 1, ..., N$, independent observations and assume that the error terms ε_i are normally and independently distributed with mean zero and variance σ^2, $[N(0, \sigma^2)]$, independent of all x_i. The density function of y_i (for a given x_i) is then given by

$$f(y_i|\beta, \sigma^2) = \frac{1}{\sqrt{(2\pi\sigma^2)}} exp\left\{ -\frac{1}{2}\frac{(y_i - \beta_1 - \beta_2 x_i)^2}{\sigma^2} \right\}$$

1.1 Give an expression for the loglikelihood contribution of observation i, $log\ L_i\ (\beta, \sigma^2)$. Explain why the log-likelihood function of the entire sample is given by

$$log\ L\ (\beta, \sigma^2) = \sum_{i=1}^{N} log\ L_i\ (\beta, \sigma^2).$$

The loglikelihood contribution is given by

$$log\ \ L_i(\beta, \sigma^2) = -\frac{1}{2} log(2\pi\sigma^2) - \frac{1}{2}\frac{(y_i - \beta_1 - \beta_2 x_i)^2}{\sigma^2},$$

that follows from the question. The error terms are independent across observations, the loglikelihood function is simply the sum of all N loglikelihood contributions. In other words, the likelihood function is identical to the joint density function of $y_1, ..., y_N$ but it is considered as a function of the unknown parameters β, σ^2.

1.2 Calculate the first derivative $\frac{\partial\ log\ L_i(\beta, \sigma^2)}{\partial \beta}$ with respect to β_1 and β_2.

Differentiation with respect to β_1 gives

$$\frac{\partial\ \ log\ \ L_i(\beta, \sigma^2)}{\partial \beta_1} = -\frac{1}{2} 2\sigma^2 \frac{(y_i - \beta_1 - \beta_2 x_i)}{\sigma^4} \times -1 = \frac{(y_i - \beta_1 - \beta_2 x_i)}{\sigma^2},$$

and similarly with respect to β_2

$$\frac{\partial\ \ log\ \ L_i(\beta, \sigma^2)}{\partial \beta_2} = -\frac{1}{2} 2\sigma^2 \frac{(y_i - \beta_1 - \beta_2 x_i)}{\sigma^4} \times -x_i = \frac{(y_i - \beta_1 - \beta_2 x_i)}{\sigma^2} x_i.$$

1.3 Suppose that x_i is a dummy variable equal to 1 for males and 0 for females, such that x_i for $i = 1, ..., N_1$ (the first N_1 observations) and $x_i = 0$ for $i = N_1 + 1, ..., N$.

Derive the first order conditions for maximum likelihood estimators for β and show that the estimators are given by:

$$\hat{\beta}_1 = \frac{1}{N-N_1}\sum_{i=N_1+1}^{N} y_i, \qquad \hat{\beta}_2 = \frac{1}{N_1}\sum_{i=1}^{N_1} y_i - \hat{\beta}_1$$

What is the interpretation of these two estimators? What is the interpretation of the true parameter values β_1 and β_2?

1) We know that the log-likelihood function is given by:

$$logL(\beta,\sigma^2) = \sum_{i=1}^{N} -\frac{1}{2}log(2\pi\sigma^2) - \frac{1}{2}\frac{(y_i-\beta_1-\beta_2 x_i)^2}{\sigma^2}.$$

2) The FOC for $\hat{\beta}_1$ is given by:

$$\sum_{i=1}^{N}(y_i-\hat{\beta}_1-\hat{\beta}_2 x_i) = 0$$

which in this case (x_i being a dummy) can be rewritten as:

$$\frac{1}{N}\sum_{i=1}^{N} y_i - \hat{\beta}_1 - \frac{N_1}{N}\hat{\beta}_2 = 0$$

3) The FOC for $\hat{\beta}_2$ is given by:

$$\sum_{i=1}^{N}(y_i-\hat{\beta}_1-\hat{\beta}_2 x_i)x_i = 0$$

and can be rewritten as follows:

$$\sum_{i=1}^{N_1}(y_i-\hat{\beta}_1-\hat{\beta}_2) = 0$$

4) Solving for $\hat{\beta}_2$ gives the ml-estimator for $\hat{\beta}_2$. Plugging $\hat{\beta}_2$ into $\hat{\beta}_1$ gives the ml-estimator for $\hat{\beta}_1$ respectively.

In this setting, β_1 corresponds to the sample average for females and β_2 is the difference in sample averages between males and females. The true parameter values correspond to the expected value for females and the expected differential in y_i between males and females.

1.4 Suppose that we are interested in the hypothesis $H_0 : \beta_2 = 0$ with alternative $H_1 : \beta_2 \neq 0$. Tests can be based upon the Likelihood ratio, Lagrange multiplier or Wald principle. Explain what these three principles are.

- The Wald test is generally applicable to any estimator that is consistent and asymptotically normal. The likelihood ratio (LR) principle provides an easy way to compare two alternative nested models, while the Lagrange multiplier (LM) tests allow one to test restrictions that are imposed in estimation. This makes the LM approach particularly suited for misspecification tests where a chosen specification of the model is tested for misspecification in heteroskedasticity, non-normality, or omitted variables.

$$\max_{\theta} \; log \; L \; (\theta) = \max_{\theta} \sum_{i=1}^{N} \; log \; L_i \; (\theta)$$

Suppose that we are interested in testing one or more linear restrictions on the parameter vector $\theta = (\theta_1, ..., \theta_K)'$. These restrictions can be summarized as $H_0 : R\theta = q$ for some fixed J-dimensional vector q, where R is a $J \times K$ matrix. Assume that the J rows of R are linearly independent, so that the restrictions are not in conflict with each other nor redundant. We summarize below these tests in detail.

- **Wald test:** Estimate θ by maximum likelihood and check whether the difference $R\hat{\theta}-q$ is close to zero, using its (asymptotic) covariance matrix. This is the idea that underlies the well-known t and F tests.
- **Likelihood ratio test:** Estimate the model twice: once without the restriction imposed (giving) and once with the null hypothesis imposed (giving the constrained maximum likelihood estimator $\hat{\theta}$, where $R\hat{\theta}=q$) and check whether the difference in loglikelihood values $log \; L(\hat{\theta})$ - $log \; L(\tilde{\theta})$ is significantly different from zero. This implies the comparison of an unrestricted and a restricted maximum of $log \; L(\theta)$.
- **Lagrange multiplier test:** Estimate the model with the restriction from the null hypothesis imposed (giving $\hat{\theta}$) and check whether the first order conditions from the general model are significantly violated. That is, check whether $\frac{\partial log L(\theta)}{\partial \theta}$ is significantly different from zero.

1.5 Discuss for each of the three tests what is required to compute them.

- **The likelihood ratio test:** We need the maximum values for the loglikelihood function when the model is estimated under the null hypothesis (the restricted model) and when the model is estimated without restrictions (the unrestricted model).
- **For the Lagrange multiplier test:** We need to estimate the model under the null, and calculate the scores of the general loglikelihood function.
- **For the Wald test:** We need the coefficient estimates (unrestricted model) and the estimated covariance matrix.

Problem 2

This problem illustrates concepts of maximum likelihood estimation in R using the example of tossing a globe in McElreath (2020).

2.1 Create fake data: Simulate realizations of a Bernoulli random variable for a sequence of $n = 100$ trials. Assume that the true proportion in the population is $\pi = 0.7$.

```
set.seed(123)
y <- rbinom( n = 100, size = 1, prob = 0.7 )
```

2.2 Defining the log-likelihood: Use the Bernoulli random variable, y_i, write the likelihood function and log-likelihood function. Provide the computer code in R.

$$L(\pi) = \prod_{i=1}^{n} \pi^{y_i}(1-\pi)^{1-y_i}$$

$$\ln L(\pi) = \sum_{i=1}^{n} y_i \ln \pi + (1-y_i)\ln(1-\pi)$$

```
log.likelihood <-  function(pi, y) {
                   sum( y*log(pi) + (1-y)*log(1-pi) )
                   }

OR

log.likelihood.a <- function(pi, y) {
                    sum( dbinom( x = y, size = 1, prob = pi, log = TRUE) )
                    }
```

2.3 Maximize the likelihood function by using Grid search in R.

```
Step 1:  Define the grid

grid <- seq( from = 0, to = 1, length.out = 5 )
grid
[1] 0.00 0.25 0.50 0.75 1.00

Step 2:  Likelihood

likelihood <- dbinom( x = sum(y), size = length(y), prob = grid)
likelihood
[1] 0.000000e+00 5.300523e-22 9.790433e-06 5.799778e-02 0.000000e+00

Step 3:  Plot

plot(grid, likelihood, type="l",
                 main="likelihood",
                 xlab="parameter")
```

Step 4: Grid value corresponding to maximum

```
grid[which.max(likelihood)]
[1] 0.75
```

Using grid search we can find the maximum of the likelihood function by trying each possible value for pi. The higher we set the value for length.out, the more precise does the grid search get. The plot shows the values of the likelihood function for different parameter values.

2.4 Do the maximization problem by using R base: optim. Compute the variance and standard errors using the inverse of the negative actual Hessian. Compare the result of the numerical optimization to the result of the gridsearch and the analytical result.

Step 1: Maximization with optim (Quasi-Newton algorithm)

```
mle <- optim( par=0.5,
              fn = log.likelihood,
              y = y,
              method = "L-BFGS-B", lower = 1e-10, upper = (1 - 1e-10),
              control = list(fnscale = -1),
              hessian = TRUE)

mle$par
[1] 0.7099993
```

Step 2: Variance and standard errors

```
round( c( (-mle$hessian)^{-1}, sqrt( (-mle$hessian)^{-1}) ), 4 )
[1] 0.0021 0.0454
```

Step 3: Compare with analytical solution

```
pi_hat <- sum(y)/length(y)
pi_hat
[1] 0.71

var_hat <- pi_hat*(1-pi_hat)/length(y)
round(var_hat,4)
[1] 0.0021
```

2.5 Do maximization by using Newton- or quasi-Newton (package maxLik). Compute the variance and the standard errors using the inverse of the negative actual Hessian.

```
library(maxLik)
mle.nr <- maxNR( fn = log.likelihood, start = c(pi=0.5), y = y )
summary(mle.nr)

--------------------------------------------
Newton-Raphson maximisation
Number of iterations: 2
Return code: 1
gradient close to zero
Function value: -60.21517
Estimates:
   estimate      gradient
pi     0.71    -2.842171e-08
--------------------------------------------
round( c( (-mle.nr$hessian)^{-1}, sqrt( (-mle.nr$hessian)^{-1}) ), 4 )
[1] 0.0021 0.0454
```

2.6 Do maximization by using BHHH algorithm. Compute the variance and standard errors using the inverse of the negative actual Hessian.

The function of the individual score contributions:

$$s_i(\pi) = \frac{dL_i(\pi)}{(d\pi)} = \frac{y_i}{\pi} - \frac{1-y_i}{1-\pi}$$

```
log.likelihood.grad <-   function(pi, y) {
                         y/pi - ( 1 - y ) / ( 1 - pi )
                         }

mle.bhhh <- maxBHHH( fn = log.likelihood,
                     grad = log.likelihood.grad,
                     start = c(pi=0.5), y = y )

summary(mle.bhhh)

BHHH maximisation
Number of iterations: 1
Return code: 1
gradient close to zero
Function value: -60.21517
Estimates:
   estimate     gradient
pi     0.71    1.487699e-14
--------------------------------------------
round( c( (-mle.bhhh$hessian)^{-1}, sqrt( (-mle.bhhh$hessian)^{-1}) ), 4 )
[1] 0.0021 0.0454
```

Problem 3

Assume that we have independent observations of a random variable y that is normally distributed with unknown mean μ and known standard deviation $\sigma = 1$, i.e. $y \sim N(\mu, 1)$. Suppose the goal is to estimate the unknown parameter μ using data on $n = 10$ realizations of the random variable y shown below.

(i)	(y)
1	-1.560476
2	-1.230177
3	0.558708
4	-0.929492
5	-0.870712
6	0.715065
7	-0.539084
8	-2.265061
9	-1.686853
10	-1.445662

3.1 Write down the likelihood function and the log likelihood function.

$$L(\mu) = (2\pi)^{-\frac{n}{2}} \prod_{i=1}^{n} \exp\left(-\frac{1}{2} \cdot (y_i - \mu)^2\right)$$

$$\ln L(\mu) = -\frac{n}{2} \ln(2\pi) - \sum_{i=1}^{n} \left(\frac{1}{2} \cdot (y_i - \mu)^2\right)$$

3.2 Derive the score function and solve the first order condition for the maximum likelihood estimator of μ.

$$s(\mu) = \sum_{i=1}^{n} (y_i - \mu) = 0$$

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

3.3 Give the Fisher information and the variance of the maximum likelihood estimator of μ.

$$I(\mu) = n$$

$$Var(\hat{\mu}) = \frac{1}{n}$$

3.4 Use the Newton-Raphson algorithm (or a quasi-Newton method) to numerically compute the maximum likelihood estimate of the unknown parameter μ. Report a 90% confidence interval. Use the data in the above table.

```
Step 1:  Function of the individual score contributions

log.likelihood <- function(mu, y) {
                  sum( dnorm(y, mean = mu, sd = 1, log = TRUE) )
                  }

Step 2:  Maximization using optim

result <- optim(
                par = 10,
                fn = log.likelihood,
                y = y,
                method = "BFGS",
                control = list(fnscale = -1),
                hessian = TRUE )

result$par
[1] -0.9253744

Step 3:  Variance  standard errors

c <- qnorm(0.95, mean = 0, sd = 1)
c( result$par - c*sqrt((-result$hessian^(-1))), result$par + c*sqrt((-result$hessian^(-1))))
[1] -1.445523 -0.405226
```

Problem 4

The dataset `vote1` in the R package `wooldridge` contains information on election outcomes and campaign expenditures for 173 two-candidate races in the U.S. (candidate A vs. candidate B). Let *voteA* be the percentage of the vote received by candidate A and *shareA* the percentage of total campaign expenditures by candidate A. We will analyze whether spending more relative to one's opponent is associated with a higher vote share.

4.1 Open `vote1` in R and make yourself familiar with the data. Have a look at the scatter plot of vote share and campaign expenditure of candidate A.

```
library(wooldridge)
data(vote1, package="wooldridge")
```

4.2 Estimate the model $voteA = \beta_0 + \beta_1 shareA + u$ using OLS. Interpret the coefficient on shareA.

```
OLS <- lm(voteA ~ shareA, data=vote1)
summary(OLS)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 26.81221    0.88721   30.22   <2e-16 ***
shareA       0.46383    0.01454   31.90   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 6.385 on 171 degrees of freedom
Multiple R-squared:  0.8561,        Adjusted R-squared:  0.8553
F-statistic:  1018 on 1 and 171 DF,  p-value: < 2.2e-16

Interpretation:  Increasing candidate A's expenditure share by one unit (1000$) is associated with
an approximate increase of 0.46 percentage points in total vote share.
```

4.3 Now compute the maximum likelihood estimates of the parameters. Compare the results obtained through maximum likelihood estimation with those from ordinary least squares.

```
Step 1:  Define log-likelihood function

loglik <- function(par, voteA) {
  x <- as.vector(vote1$shareA)
  sum(dnorm(voteA, mean = par[1] + par[2] * x, sd = par[3], log = TRUE))
}

Step 2:  Maximization via optim

MLE <- optim(c(intercept = 20, shareA = 0.5, sd = 1),
             fn = loglik,
             voteA = as.vector(vote1$voteA),
             control = list(fnscale = -1),
             hessian = TRUE)

Step 3:  Calculate standard errors

se <- round(sqrt(diag(solve(-MLE$hessian))), 4)

Step 4:  Compile results

results <- data.frame(
  Estimate = MLE$par[1:2],
  Standard_Error = se[1:2]
)
print(results)

          Estimate Standard_Error
intercept 26.81244         0.8822
shareA     0.46385         0.0145
```

References

McElreath, R. (2020). *Statistical rethinking- A Bayesian course with examples in R and Stan*. CRC press

Verbeek, M. (2012). *A Guide to Modern Econometrics*. John Wiley & Sons, Chichester.

Wooldridge, J. (2013). *Introductory Econometrics. A modern approach*.