Assignment 2 - CSC/DSC 265/465 - Spring 2019 - SOLUTIONS

Unless otherwise specified, statistical significance can be accepted when the relevant P-value is no larger than $\alpha = 0.05$. Note that problem **Q5** is reserved for graduate students.

Q1: Suppose, conditional on parameter $\theta \in (0,1)$, a random variable X has distribution $X \sim bin(n,\theta)$. Then suppose we assign a prior distribution $\pi(\theta)$ to θ of the form $\pi(1/4) = \pi(1/2) = \pi(3/4) = 1/3$. If $n = 10$ and we observe $X = 4$, give the posterior distribution $\pi(\theta \mid X)$ of θ.

SOLUTION: Interpreting X as having a binomial distribution conditional on θ, we write

$$P(X = x \mid \theta) = \binom{n}{x}\theta^x(1-\theta)^{n-x},$$

leading to posterior distribution

$$\begin{aligned}\pi(\theta \mid X = x) &= \frac{P(X = x \mid \theta)\pi(\theta)}{P(X = x)} \\ &= \frac{P(X = x \mid \theta)\pi(\theta)}{\sum_{\theta \in \{1/4,1/2,3/4\}} P(X = x \mid \theta)\pi(\theta)}\end{aligned}$$

For $x = 4$, we have

$$\begin{aligned}P(X = 4 \mid \theta = 1/4)\pi(1/4) &= (1/3) \times \tbinom{10}{4}(1/4)^4(1-1/4)^6 = (1/3) \times \tbinom{10}{4} \times 3^6/4^{10} \\ P(X = 4 \mid \theta = 1/2)\pi(1/2) &= (1/3) \times \tbinom{10}{4}(2/4)^4(1-2/4)^6 = (1/3) \times \tbinom{10}{4} \times 2^{10}/4^{10} \\ P(X = 4 \mid \theta = 3/4)\pi(3/4) &= (1/3) \times \tbinom{10}{4}(3/4)^4(1-3/4)^6 = (1/3) \times \tbinom{10}{4} \times \times 3^4/4^{10}.\end{aligned}$$

Note that all factors common to each expression will cancel when evaluating the posterior distribution. This means

$$\begin{aligned}\pi(\theta = 1/4 \mid X = 4)\pi(1/4) &= \tfrac{3^6}{3^6+2^{10}+3^4} \approx 0.397 \\ \pi(\theta = 1/2 \mid X = 4)\pi(1/4) &= \tfrac{2^{10}}{3^6+2^{10}+3^4} \approx 0.558 \\ \pi(\theta = 3/4 \mid X = 4)\pi(1/4) &= \tfrac{3^4}{3^6+2^{10}+3^4} \approx 0.044.\end{aligned}$$

Q2: Suppose the traffic flow rate on a certain highway is expressed as λ vehicles per hour. In other words, the number of vehicles which pass a given point in one hour is, on average, λ.

Then, suppose at one point on this highway there is a toll booth installation with a maximum capacity of $N = 10$ toll booths. The number of tolls currently open is $M \leq N$. When a car approaches the installation, any one of the open tolls is chosen at random. As an approximation, we will assume that the number of vehicles which pass through a given open toll in a time interval of length T has a Poisson distribution with mean $\lambda T/M$. We may also assume that the number of vehicles passing through distinct tolls are statistically independent.

We will construct a Bayesian model for the estimation of λ. Both λ and M are unknown, and have (independent) prior distributions $\pi_\lambda(\lambda)$, $\pi_M(M)$. The observation will be a low resolution "snapshot" of the toll booth installation. Within a period of $T = 15$ seconds, X is the number of toll booths through which at least one vehicle has passed.

(a) What is $f(x \mid \lambda, M)$, the distribution of X conditional on (λ, M)?

(b) Give an expression for the posterior distribution of the parameter pair (λ, M).

(c) Create a Hastings-Metropolic MCMC algorithm to simulate a sample from the posterior distribution of (λ, M). Include the following elements:

(i) The prior distribution of λ is uniform.

(ii) The prior distribution of M is given by $\pi_M(8) = 4/7$, $\pi_M(9) = 2/7$, $\pi_M(10) = 1/7$, with $\pi_M(m) = 0$ for any $m < 8$.

(iii) A proposal distribution generates proposed state (λ_{new}, M_{new}) from (λ_{old}, M_{old}) in the following way. Set

$$\lambda_{new} = \lambda_{old} + U,$$

where $U \sim unif(-10, 10)$. If $M_{old} = 8$ or $M_{old} = 10$ set $M_{new} = 9$ with probability one. If $M_{old} = 9$ set $M_{new} = 8$ or $M_{new} = 10$ with equal probabilty. Note that a proposed state is generated for both parameters λ and M within each transition.

(iv) Allow the MCMC to run for 5,000,000 transitions. However, store the current values of λ M only at intervals of 1,000 transitions.

(v) Use $M_{old} = 8$ and $\lambda_{old} = X \times 240$ as initital states, for observation X.

Run the MCMC sampler four times, using observations $X = 2, 4, 6, 8$.

(d) For each of the four samples, give an estimate of marginal posterior distribution of M. Compare this to the prior distribution π_M. How do the posterior distributions of M vary with observations $X = 2, 4, 6, 8$?

(e) Create side by side boxplots of the sampled λ values for observations $X = 2, 4, 6, 8$. How do the posterior distributions of λ vary with observations $X = 2, 4, 6, 8$?

SOLUTION:

(a) For a Poisson random variable Y with mean τ, we have $P(Y = y) = \tau^y e^{-\tau}/y!$, $y = 0, 1, 2, \ldots$. In particular, $P(Y = 0) = e^{-\tau}$. The number of vehicles passing through a single toll booth is a Poisson random variable with mean $\tau = \lambda T/M$, when λ, M are fixed. So, conditional on λ, M, we have $X \sim bin(n, p)$, where $n = M$, $p = (1 - e^{-\lambda T/M})$. Since λ is in units of vehicles per hour, we have $T = 15$ sec $= 1/240$ hours. This gives

$$f(x \mid \lambda, M) = \binom{M}{x} (1 - e^{-\lambda/(240 \times M)})^x (e^{-\lambda/(240 \times M)})^{M-x}.$$

(b) From Bayes rule:

$$\begin{aligned} \pi(\lambda, M \mid X = x) &= \frac{f(x \mid \lambda, M)\pi_\lambda(\lambda)\pi_M(M)}{P(X = x)} \\ &\propto \binom{M}{x} (1 - e^{-\lambda/(240 \times M)})^x (e^{-\lambda/(240 \times M)})^{M-x} \times \pi_M(M) \times \pi_\lambda(\lambda). \end{aligned}$$

We only need to specify the posterior density up to a constant of proportionality.

(c) The following code implements the MCMC algorithm:

```
### P(X | lambda, M)

p0 = function(lambda,m,x) { dbinom(x,size=m,prob=(1-exp(-lambda/(m*240))),log=TRUE ) }

### Prior for M

pim = c(0,0,0,0,0,0,0,4,2,1)
pim = pim/sum(pim)

### set up MCMC algorithm

# Do 5,000,000 transitions, store at intervals of 1,000 transitions

niter = 5000000
ntrace = niter/1000

# Set random seed

set.seed(234)

# We'll do 4 samples, one for each of X = 2,4,6,8

x.vector  = c(2,4,6,8)

lambda.trace.list = list()
m.trace.list = list()

###
### Loop through X = 2,4,6,8
###

for (iii in 1:4) {

  lambda.trace = rep(NA,ntrace)
  m.trace = rep(NA,ntrace)

  ###
  ### Loop through iterations
  ###

  # Set data value X

  x = xvec[iii]

  # Set initial values for M and lambda

  m.old = 8
  lambda.old = x*240

  j = 0
  for (i in 1:niter) {

    # Simulate transition from proposal distribution
```

```
    lambda.new = lambda.old + runif(1,min=-10,max=10)
    lambda.new = max(0,lambda.new)

    if (m.old == 10) {m.new = 9}
    if (m.old == 9) {m.new = m.old + sample(c(-1,1),size = 1)}
    if (m.old == 8) {m.new = 9}

    if (m.new == 10) {bal = 2}
    if (m.new == 9) {bal = 1/2}
    if (m.new == 8)  {bal = 2}

    # Determine acceptance probability (the q.ratio is always 1)

    alpha = p0(lambda.new,m.new,x) + log(pim[m.new]) - p0(lambda.old,m.old,x) - log(pim[m.old])
    alpha = exp(alpha)
    alpha = alpha*bal

    # Accept or reject proposal

    if (runif(1) <= alpha) {
      lambda.old = lambda.new
      m.old = m.new
    }

    # capture samples

    if (i %% 1000 == 0) {
      j = j+1
      lambda.trace[j]=lambda.old
      m.trace[j]=m.old
    }
  }

  # Store sample for current value of X

  lambda.trace.list[[iii]] = lambda.trace
  m.trace.list[[iii]] = m.trace

}
```

(d) Marginal posterior distributions of M and λ can be obtained directly from the sample. The following code summarizes the estimated marginal posterior distributions of M.

```
> ###
> ### Create and display posterior distributions of M
> ###
>
> f.msum = function(x) {
+   c(mean(x==8),mean(x==9),mean(x==10))
+ }
>
> msum = sapply(m.trace.list, f.msum)
> msum = cbind(c(4,2,1)/7,msum)
> rownames(msum) = paste('M=',8:10)
> colnames(msum) = c('Prior',paste('X=',xvec))
>
> msum
```

```
          Prior   X= 2   X= 4   X= 6   X= 8
M= 8  0.5714286 0.5734 0.6046 0.6108 0.7216
M= 9  0.2857143 0.2870 0.2660 0.2708 0.2120
M= 10 0.1428571 0.1396 0.1294 0.1184 0.0664
>
```

As X increases, the marginal posterior distributions of M become more concentrated around $M = 8$. That is, larger values of X provide greater evidence that $M = 8$. See table above.

(e) The following code summarizes the estimated marginal posterior distributions of λ. Clearly, as X increases, the Bayesian estimates of λ increase also. See Figure 1.

```
###
### Posterior distributions of lambda
###

boxplot(lambda.trace.list,names = paste('X= ',c(2,4,6,8),sep=""),
        ylab='Posterior distribution of lambda')
```

Q3: This problem will make use of the `Cars93` data set from the `MASS` library, titled **Data from 93 Cars on Sale in the USA in 1993**. First, select a subset of variables from `Cars93` with the following code:

```
> carsb = Cars93[,c(4,5,6,7,8,12,13,14,15,17,19:22,25,26)]
> names(carsb)
 [1] "Min.Price"          "Price"
 [3] "Max.Price"          "MPG.city"
 [5] "MPG.highway"        "EngineSize"
 [7] "Horsepower"         "RPM"
 [9] "Rev.per.mile"       "Fuel.tank.capacity"
[11] "Length"             "Wheelbase"
[13] "Width"              "Turn.circle"
[15] "Weight"             "Origin"
>
```

The first 15 columns of `carsb` are continuously quantitative automobile features. Column 16 is the factor `Origin` possessing two levels `USA`, `non-USA`, indicating whether or not the manufacturer is located in the USA. Do a log transformation of the first 15 columns:

```
> carsb[,-16] = log(carsb[,-16])
```

The objective will be to build a classifier of `Origin` based on the remaining 15 quantitative features. The class to be predicted will be either `USA` or `non-USA`.

	true `USA`	true `non-USA`
predicted `USA`	n_{11}	n_{12}
predicted `non-USA`	n_{21}	n_{22}

Any record used for testing the predictor is placed in exactly one of the four cells.

(a) Recall the odds representation of Baye's Rule, in this application:

$$\begin{aligned} Odds(\text{true } \texttt{USA} \mid \text{predicted } \texttt{USA}) &= LR_+ \times Odds(\text{true } \texttt{USA}) \\ Odds(\text{true } \texttt{USA} \mid \text{predicted } \texttt{non-USA}) &= LR_+ \times Odds(\text{true } \texttt{USA}) \end{aligned}$$

Express LR_+ and LR_- in terms of the elements $(n_{11}, n_{12}, n_{21}, n_{22})$ of the confusion table. Create an `R` function that inputs the confusion table, and outputs a single vector with elements (CE, LR_+, LR_-), where CE is classification error.

(b) Using the function `lda()` fit a classifier using linear discriminant analysis (LDA). Use the function of Part (a) to record (CE, LR_+, LR_-). Do not use cross-validation to fit this classifier.

(c) Using the function `qda()` fit a classifier using quadratic discriminant analysis (QDA). Use the function of Part (a) to record (CE, LR_+, LR_-). Do not use cross-validation to fit this classifier. Does the QDA classifier appear to improve the LDA classifier?

(d) Repeat Parts (b) and (c) using the `CV = TRUE` option of the `lda()` and `qda()` functions. This yields the predictions resulting from *leave-one-out* cross-validation. How does this change the comparison between the LDA and QDA classifier. Give a brief explanation.

SOLUTION:

(a) The Baye's Rule using odds is:

$$Odds(A \mid E) = \frac{P(E \mid A)}{P(E \mid A^c)} \times Odds(A).$$

Therefore, if

$$Odds(\text{true USA} \mid \text{predicted USA}) = LR_+ \times Odds(\text{true USA})$$

we have

$$LR_+ = \frac{P(\text{predicted USA} \mid \text{true USA})}{P(\text{predicted USA} \mid \text{true non-USA})} = \frac{n_{22}/(n_{12}+n_{22})}{n_{21}/(n_{11}+n_{21})}.$$

Similarly, if

$$Odds(\text{true USA} \mid \text{predicted non-USA}) = LR_- \times Odds(\text{true USA})$$

we have

$$LR_- = \frac{P(\text{predicted non-USA} \mid \text{true USA})}{P(\text{predicted non-USA} \mid \text{true non-USA})} = \frac{n_{12}/(n_{12}+n_{22})}{n_{11}/(n_{11}+n_{21})}.$$

The following code creates the required functions:

```
library(MASS)
library(class)

lrfp = function(m) { (m[2,2]/(m[1,2]+m[2,2]))/(m[2,1]/(m[1,1]+m[2,1]))  }
lrfn = function(m) { (m[1,2]/(m[1,2]+m[2,2]))/(m[1,1]/(m[1,1]+m[2,1]))  }
f0 = function(m) {
  x = c(1-sum(diag(m))/sum(m),lrfp(m),lrfn(m))
  names(x) = c("CE","LR.pos","LR.neg")
  return(x)
}
```

(b) The following code may be used.

```
> fit.lda= lda(Origin~., data=carsb)
> confusion.matrix.lda = table(predict(fit.lda)$class,carsb$Origin)
> confusion.matrix.lda

          USA non-USA
  USA      43       5
  non-USA   5      40
> f0(confusion.matrix.lda)
       CE    LR.pos    LR.neg
0.1075269 8.5333333 0.1240310
```

(c) The following code may be used. For the QDA classifier CE is smaller, LR_+ is larger and LR_- is smaller. By all three measures, the QDA classifier is more accurate.

```
> fit.qda= qda(Origin~., data=carsb)
> confusion.matrix.qda = table(predict(fit.qda)$class,carsb$Origin)
> confusion.matrix.qda

          USA non-USA
  USA      46       1
  non-USA   2      44
> f0(confusion.matrix.qda)
         CE      LR.pos      LR.neg
 0.03225806 23.46666667  0.02318841
```

(d) The following code may be used. In contrast with Part (c), by all three measures, the LDA classifier is more accurate after cross-validating. The QDA method fits a separate covariance matrix for each class, and therefore has more parameters. This makes QDA more susceptible to over-fitting, which would be corrected by cross-validation.

```
> fit.lda= lda(Origin~., data=carsb, CV=TRUE)
> confusion.matrix.lda = table(fit.lda$class,carsb$Origin)
> confusion.matrix.lda

          USA non-USA
  USA      41       6
  non-USA   7      39
> f0(confusion.matrix.lda)
       CE    LR.pos    LR.neg
0.1397849 5.9428571 0.1560976
>
> fit.qda= qda(Origin~., data=carsb, CV=TRUE)
> confusion.matrix.qda = table(fit.qda$class,carsb$Origin)
> confusion.matrix.qda

          USA non-USA
  USA      36      11
  non-USA  12      34
> f0(confusion.matrix.qda)
       CE    LR.pos    LR.neg
0.2473118 3.0222222 0.3259259
```

Q4: This problem will make use of the `Pima.tr` data set from the `MASS` library. From the `help` page:

> A population of women who were at least 21 years old, of Pima Indian heritage and living near Phoenix, Arizona, was tested for diabetes according to World Health Organization criteria. The data were collected by the US National Institute of Diabetes and Digestive and Kidney Diseases.

Note that `Pima.tr` is a subset of the complete data set containg $n = 200$ records. First, select a subset of variables from `Pima.tr` with the following code:

```
> pima1 = rbind(Pima.tr)[,c(2,3,4,5,6,8)]
> names(pima1)
[1] "glu"  "bp"   "skin" "bmi"  "ped"  "type"
>
```

The first 5 columns of `pima1` are continuously quantitative automobile features. Column 6 is the factor `type` possessing two levels `Yes` (diabetic according to WHO criteria) and `No`. The objective will be to build a classifier of `type` (ie. to predict diabetes) based on the remaining 5 quantitative features.

(a) Determine the frequencies of the `Yes` and `No` outcomes. If a classifier simply predicted the most frequent outcome for all inputed features, what would the classification error be (assuming the outcome prevalences found in the data)?

(b) Create an `R` function which accepts a vector `k.list` of values of K (the neighborhood size of the classifier), a training set of features, and a paired training set of classes. For each K in `k.list` a KNN fit will be evaluated using LOOCV. Use the `knn.cv()` function from the library `class`. The function should output the summary (CE, LR_+, LR_-) for each K in `k.list`.

(c) Apply the function of Part (b) to the data set constructed in Part (a). Use `k.list = seq(1,125,2)`. What advantage is there to using only odd numbers for K? Plot the estimated CE against K. Superimpose on the plot a horizontile line representive the value of CE estimated for the simple classifier described in Part (a). What is the minimum value of CE, and for what value of K is this attained? Give also the maximum and minimum LR_+ and LR_- attained for any K.

(d) Create side-by-side boxplots of the 5 features (make sure you use a common vertical axis). Then normalize each of the 5 features by subtracting the mean, then dividing by the standard deviation. Repeat Part (c). What difference do you notice? Given that the default metric used to define the neighborhood is Euclidean distance, why might normalizing the features improve the classifier?

SOLUTION:

(a) Use the following code to obtain the frequencies:

```
> table(pima1$type)

 No Yes
132  68
>
```

The most frequent class is `No`, with a frequency of $132/(132 + 68) = 0.66$. If a classifier simply predicts `No`, regardless of any input data, it will be correct 66% of the time, giving a classification error of $CE = 1 - 0.66 = 0.34$.

(b) The following function performs the required tasks (using function `f0` from Question 3).

```
knn.function = function(k.list, xtrain, gr) {

  pr.tab = matrix(NA,length(k.list),3)
```

```
  for (i in 1:length(k.list)) {
    knn.fit = knn.cv(xtrain,gr,k=k.list[i],use.all=T)
    cm = table(knn.fit,gr)
    pr.tab[i,] = f0(cm)
  }
  return(pr.tab)
}
```

(c) The following code can be used.

```
> # Calculate KNN classifiers
>
> k.list = seq(1,125,2)
> cv.tab = knn.function(k.list,pima1[,1:(n-1)],pima1[,n])
>
> # Plot CE vs K, superimpose baseling CE from Part (a)
>
> nyes = sum(pima1$type=='Yes')
> pdf("figA2Q4c.pdf")
> par(mfrow=c(1,1))
> plot(k.list,cv.tab[,1],ylab='CE',xlab="K")
> abline(h=nyes/ns,col='gray')
> dev.off()
RStudioGD
        2
>
> # Get best accuracy measures
>
> min(cv.tab[,1])
[1] 0.24
> which.min(cv.tab[,1])
[1] 16
> max(cv.tab[,2],na.rm=T)
[1] 4.714286
> min(cv.tab[,3])
[1] 0.5546218
>
```

Since there are two classes, if K is an even number, we may have ties when making a classification. Using only odd numbers for K avoids this problem. See Figure 2 for plot. From the output above, the minimum classification error is $CE = 0.24$ for $K = 16$. The maximum LR_+ is 4.714, and the minimum LR_- is 0.555.

(d) The following code can be used.

```
  # Create boxplot
>
> pdf("figA2Q4d1.pdf")
> boxplot(pima1[,-6])
> dev.off()
RStudioGD
        2
>
> # Renormalize
>
> for (i in 1:(n-1)) {pima1[,i] = (pima1[,i] - mean(pima1[,i]))/sd(pima1[,i]) }
>
```

```
> ### Redo analysis of Part (c)
>
> # Calculate KNN classifiers
>
> k.list = seq(1,125,2)
> cv.tab = knn.function(k.list,pima1[,1:(n-1)],pima1[,n])
>
> # Plot CE vs K, superimpose baseling CE from Part (a)
>
> nyes = sum(pima1$type=='Yes')
> pdf("figA2Q4d2.pdf")
> par(mfrow=c(1,1))
> plot(k.list,cv.tab[,1],ylab='CE',xlab="K")
> abline(h=nyes/ns,col='gray')
> dev.off()
RStudioGD
        2
>
> # Get best accuracy measures
>
> min(cv.tab[,1])
[1] 0.22
> which.min(cv.tab[,1])
[1] 18
> max(cv.tab[,2],na.rm=T)
[1] 10.67647
> min(cv.tab[,3])
[1] 0.5246423
>
```

See Figure 3 for the boxplot, and Figure 4 for the CE plot. From the output above, the minimum classification error is $CE = 0.22$ for $K = 18$. The maximum LR_+ is 10.676, and the minimum LR_- is 0.525. By all three measures the new classifier is more accurate. Examining Figure 3, we can see that the five features differ considerably by variance. Since unweighted Euclidean distance is used to define a neighborhood, features with smaller variance will have less influence on the prediction. By normalizing the features, this effect will be eliminated.

Q5: [For Graduate Students] We will revist Question 2. To simplify the problem we will assume that M is fixed and known to be $M = 10$.

(a) Derive the log-likelihood function of λ for given observation $X = x$. Plot this function against λ for $X = 6$.

(b) Give a closed form expression for the maximum likelihood estimate (MLE) of λ. Create a table of the MLEs for $x = 0, 1, \ldots, 9$.

(c) What happens when $x = 10$? Is it possible to give a MLE for λ, of any kind, for this case?

SOLUTION:

(a) We can write the log-likelihood function

$$\ell(\lambda; x) = x \log(1 - e^{-\lambda T/M}) - (M - x)\lambda T/M,$$

ignoring terms which do not depend on λ. The log-likelihood function can be plotted with the following code. See Figure 5.

```
x = 6
m = 10
p0 = function(lambda,m,x) { dbinom(x,size=m,prob=(1-exp(-lambda/(m*240))),log=TRUE )  }

lg = seq(1,10000,1)
pdf('figA2Q5a-2020.pdf')
plot(p0(lg,m,x),type='l',xlab = 'lambda', ylab = 'log-likelihood')
dev.off()
```

(b) First, note that if $x = 0$, the log-likelihood is

$$\ell(\lambda; x) = -T\lambda,$$

which is maximized over $\lambda \in [0, \infty)$ by $\lambda = 0$. That is, when $x = 0$, $\hat{\lambda}_{mle} = 0$.

Then assume $x > 0$. The derivative of the log-likelihood with respect to λ is

$$\begin{aligned}\frac{\partial \ell(\lambda; x)}{\partial \lambda} &= x \frac{(T/M)e^{-\lambda T/M}}{1 - e^{-\lambda T/M}} - (M - x)T/M \\ &= x \frac{(T/M)}{e^{\lambda T/M} - 1} - (M - x)T/M.\end{aligned}$$

After setting the partial derivative equal to zero, we can solve for λ directly.

$$x \frac{(T/M)}{e^{\lambda T/M} - 1} = (M - x)T/M,$$

which simplifies to

$$\frac{1}{e^{\lambda T/M} - 1} = (M - x)/x,$$

then

$$e^{\lambda T/M} - 1 = \frac{x}{M - x},$$

then finally

$$\hat{\lambda}_{mle} = -\frac{M}{T} \log(1 - x/M).$$

To confirm that this solution is a maximum, evalute the second derivative:

$$\begin{aligned}\frac{\partial^2 \ell(\lambda; x)}{\partial \lambda^2} &= x\frac{(T/M)^2 e^{\lambda T/M}}{(e^{\lambda T/M}-1)^2} \\ &= -x\frac{(T/M)^2 e^{-\lambda T/M}}{(1-e^{-\lambda T/M})^2} \\ &< 0.\end{aligned}$$

The second derivative is strictly negative, so the log-likelihood is strictly concave. This means the stationary point is a global maximum. Note that the expression $\hat{\lambda}_{mle} = -\frac{M}{T}\log(1 - x/M)$ also holds for $x = 0$.

The following code creates the required table.

```
> xvec = 0:9
> mle = -(m*240)*log(1 - xvec/m)
> mle.tab = cbind(xvec,mle)
> colnames(mle.tab) = c('x','mle')
> mle.tab
      x       mle
 [1,] 0    0.0000
 [2,] 1  252.8652
 [3,] 2  535.5445
 [4,] 3  856.0199
 [5,] 4 1225.9815
 [6,] 5 1663.5532
 [7,] 6 2199.0978
 [8,] 7 2889.5347
 [9,] 8 3862.6510
[10,] 9 5526.2042
```

(c) If we substitute $x = 10$ into the expression for $\hat{\lambda}_{mle}$, we get $\hat{\lambda}_{mle} = \infty$ (taking $\log(0)$ to equal $-\infty$). This means that the probability that no cars pass through an individual toll booth within 15 seconds is $P(Y = 0) = e^{-\lambda T/M} = 0$. In otherwords, the inference is that cars pass through the toll booths more or less constantly.

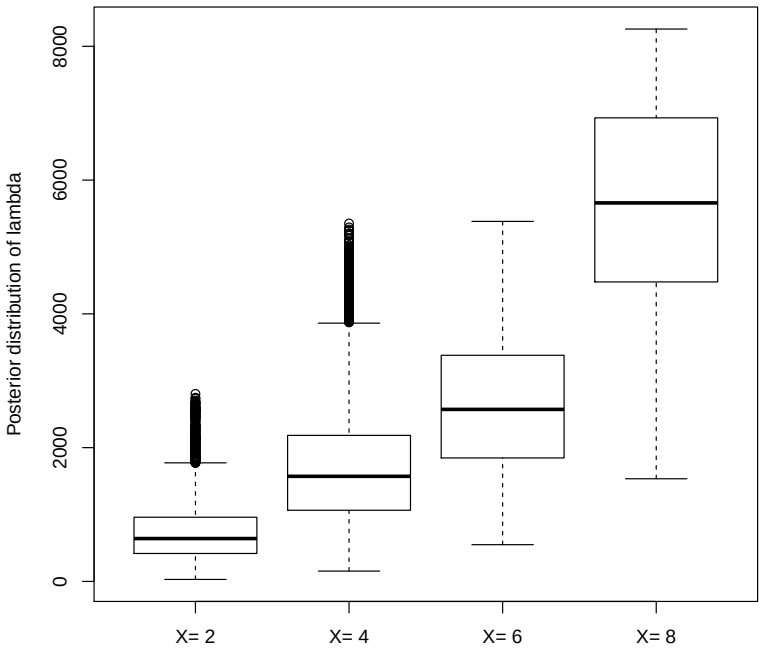

Figure 1: Figure for Q2 (e).

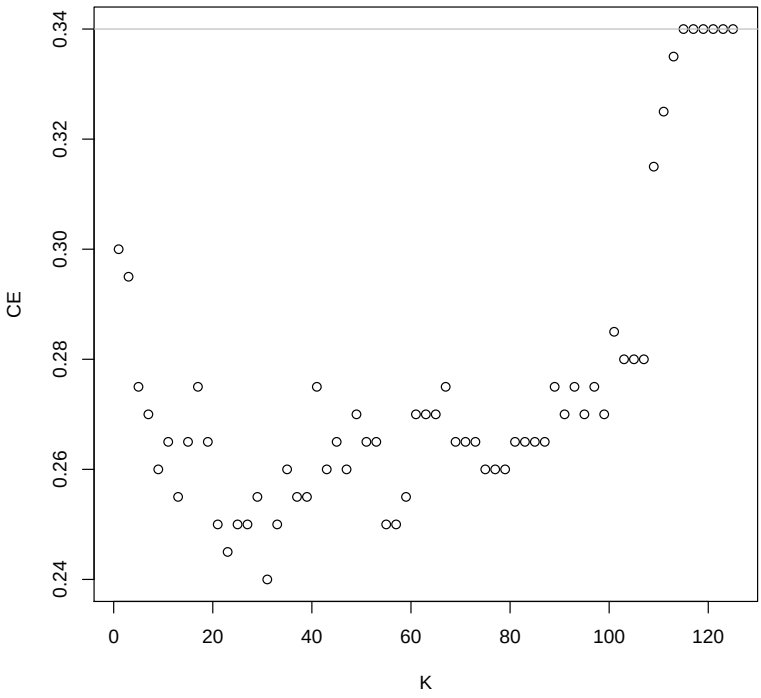

Figure 2: Figure for Q4 (c).

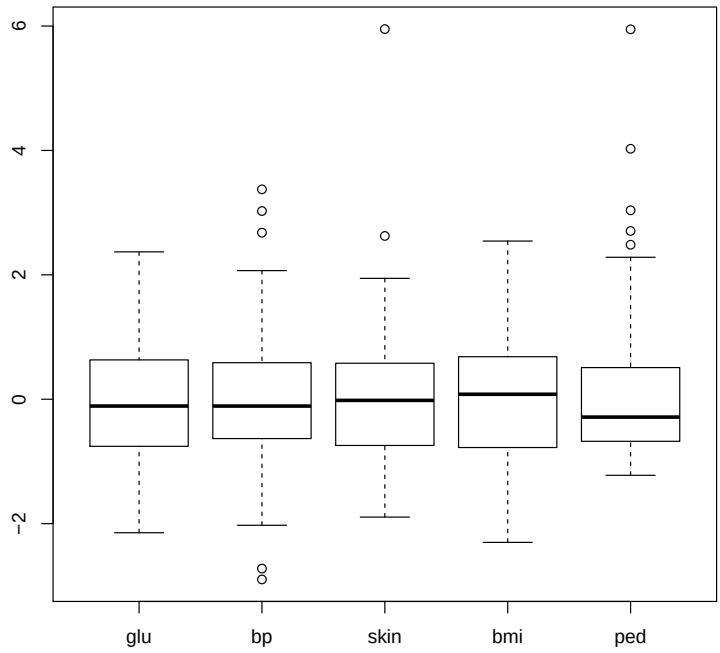

Figure 3: Figure for Q4 (d) [1/2].

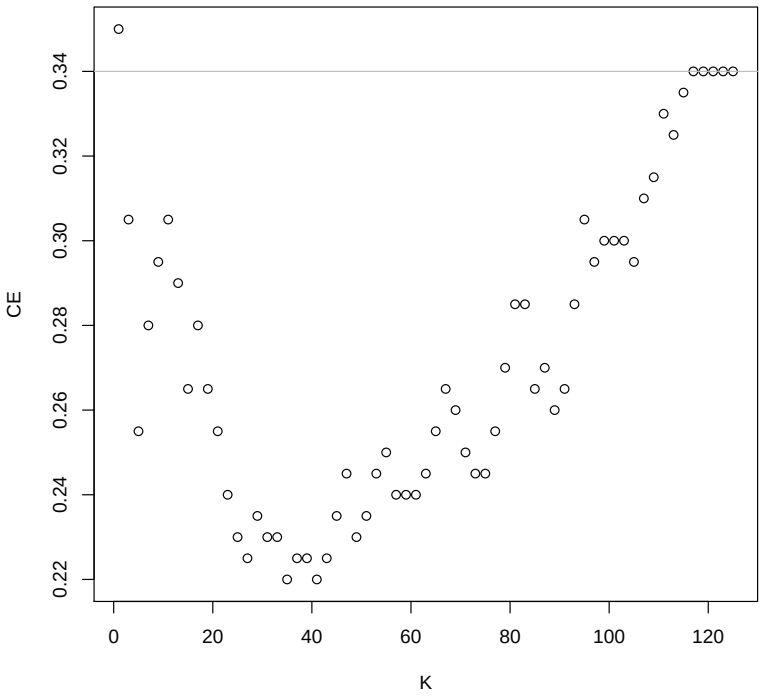

Figure 4: Figure for Q4 (d) [2/2].

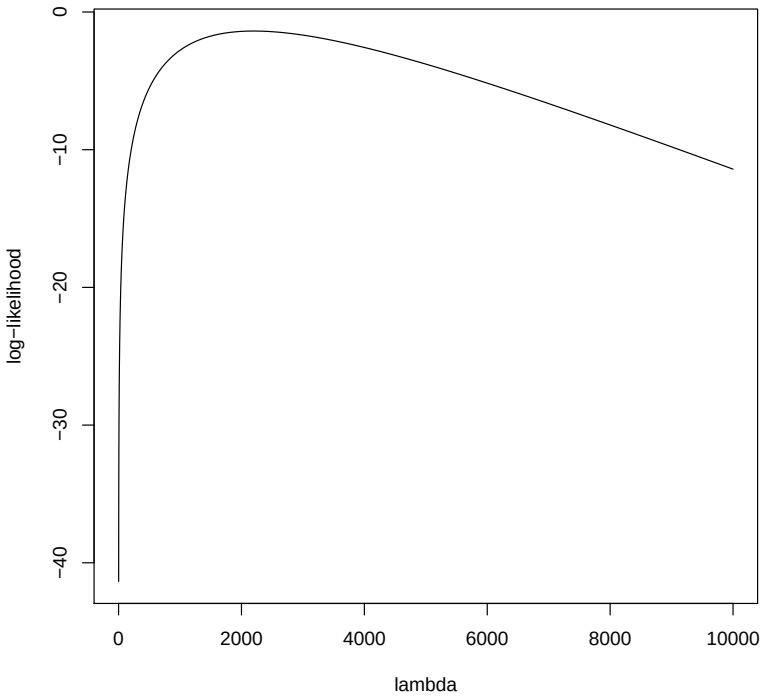

Figure 5: Figure for Q5 (a).