```
options(prompt="> ", continue="  ")
```

THE UNIVERSITY OF AUCKLAND

TERM TEST - SEMESTER 1, 2019
Campus: City

STATISTICS

Statistical Computing

(Time allowed: 50 Minutes)

INSTRUCTIONS

- Attempt ALL questions.
- Total marks are 40.
- Calculators are permitted.

1. **Write down the evaluation results** of the following R expressions, which are not necessarily meaningful in practice. Each result is worth 2 marks.

(a)
```
c(1, 2, "three")

## [1] "1"     "2"     "three"
```

(b)
```
(1:10)^2

##  [1]   1   4   9  16  25  36  49  64  81 100
```

(c)
```
matrix(1:6, nrow=2)

##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6
```

(d)
```
switch("a", b=1, c=2, 3)

## [1] 3
```

(e)
```
factor(c("male", "female"))

## [1] male   female
## Levels: female male
```

[10 marks]

2. This question makes use of data on 10,000 electric scooter trips in Austin, Texas, USA. Some summaries of the data are shown below.

```
dockless <- read.csv("AustinDockless.csv")
```

```
dim(dockless)

## [1] 10000    20

names(dockless)
```

```
##  [1] "ID"                        "Device.ID"
##  [3] "Vehicle.Type"              "Trip.Duration"
##  [5] "Trip.Distance"             "Start.Time"
##  [7] "End.Time"                  "Modified.Date"
##  [9] "Month"                     "Hour"
## [11] "Day.of.Week"               "Council.District..Start."
## [13] "Council.District..End."    "Origin.Cell.ID"
## [15] "Destination.Cell.ID"       "Year"
## [17] "Start.Latitude"            "Start.Longitude"
## [19] "End.Latitude"              "End.Longitude"
```

```r
head(dockless[c(4, 5, 10)])

##   Trip.Duration Trip.Distance Hour
## 1          1764          5214   21
## 2          1164          1512   22
## 3          1165          5359   22
## 4          2394          4473   21
## 5           101           144   22
## 6           677          1319   22
```

`Trip.Duration` is in seconds and `Trip.Distance` is in metres.

Write R expressions to calculate the following values (the correct output is shown just to give an indication of the type of output your code should produce):

(a) The number of trips longer than 1 hour (3600 seconds)

[2 marks]

```r
sum(dockless$Trip.Duration > 3600)

## [1] 120
```

(b) The hour, distance, and duration of the longest trip (in terms of distance)

[4 marks]

```r
subset(dockless, Trip.Distance == max(Trip.Distance), c(10, 5, 4))

##      Hour Trip.Distance Trip.Duration
## 4669   13         19355          2974
```

(c) The average distance for short trips versus long trips (where a short trip is less than 10 minutes)

[4 marks]

```r
with(dockless,
     tapply(Trip.Distance, cut(Trip.Duration, c(0, 600, Inf)), mean))

##   (0,600] (600,Inf]
##  751.5709 2842.0523
```

[10 marks]

3. This question also makes use of the data on 10,000 electric scooter trips in Austin, Texas, USA.

    (a) We will work only with the positive `Trip.Duration` values.
    Write code to extract just these values into a new vector called `durations`.
    [2 marks]

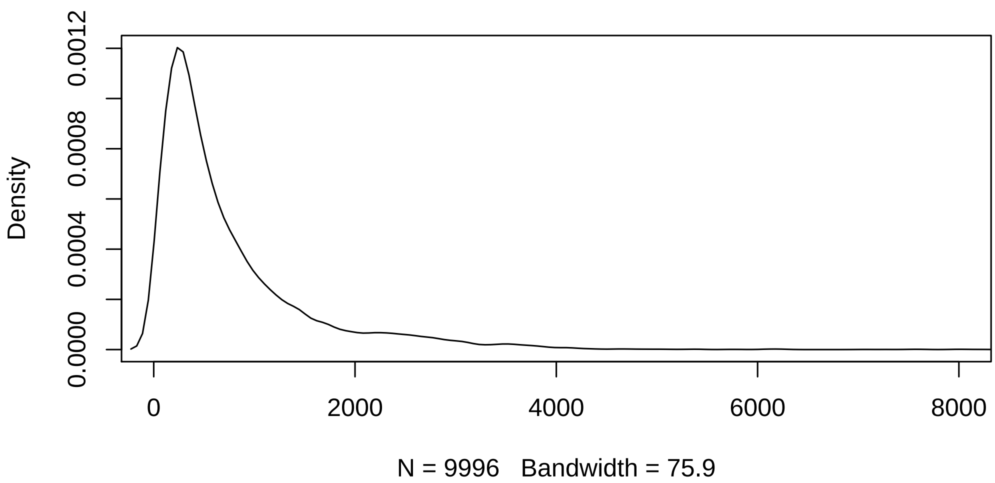
```
durations <- dockless$Trip.Duration[dockless$Trip.Duration > 0]
```

The following code and output shows how the `durations` variable relates to the `Trip.Duration` variable.

```
length(dockless$Trip.Duration)

## [1] 10000

sum(dockless$Trip.Duration <= 0)

## [1] 4

length(durations)

## [1] 9996
```

The plot below shows the distribution of the `durations` variable.

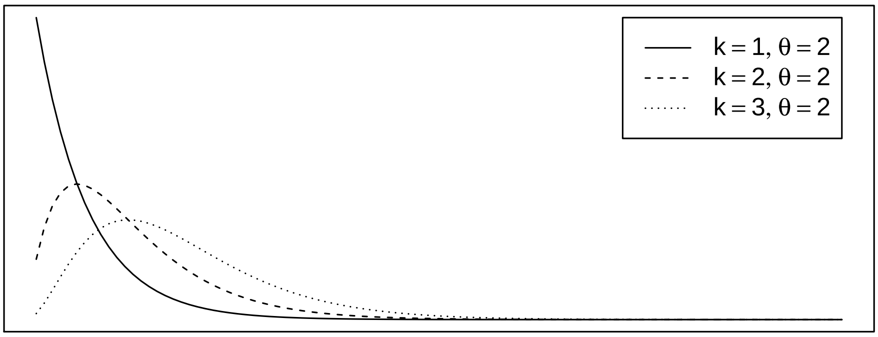
```
plot(density(durations), main="", zero.line=FALSE, xlim=c(0, 8000))
```



We assume that the `Trip.Durations` are i.i.d. Gamma(k, θ) and we want to estimate values for the parameters k and θ using Maximum Likelihood.

The plot below shows the gamma distribution for different values of k and θ.

```
t <- seq(0.1, 8, length=101)
g <- cbind(dgamma(t, 1, 2),
           dgamma(t, 2, 2),
           dgamma(t, 3, 2))
plot.new()
plot.window(range(t), range(g))
lines(t, g[,1], col="black")
lines(t, g[,2], lty="dashed")
lines(t, g[,3], lty="dotted")
box()
legend(max(t), max(g), xjust=1,
       expression(list(k == 1, theta == 2),
                  list(k == 2, theta == 2),
                  list(k == 3, theta == 2)),
       lty=c("solid", "dashed", "dotted"))
```



The log-likelihood is

$$\sum_{i=1}^{n} \log f(x_i; k, \theta)$$

where $f(x; k, \theta)$ is the gamma probability density function

$$\frac{1}{\Gamma(k)\, \theta^k}\, x^{k-1} e^{-x/\theta}$$

for $0 < \theta < \infty$, $0 < k < \infty$, and $0 \leq x < \infty$.

(b) **Write a function** `loglike()` to calculate the log-likelihood. This function should have a single argument `theta` which should be a vector of two values corresponding to the two parameters of the gamma distribution (k and θ).

[4 marks]

NOTE: The R function `dgamma()` calculates the probability density function for the gamma distribution, just like the `dnorm()` does for the Normal distribution. The `dgamma()` function also has a `log` argument just like the `dnorm()` function.

```
loglike <- function(theta) {
    sum(dgamma(durations, theta[1], theta[2], log=TRUE))
}
```

The following code and output provides some examples of how your function would be used.

```
loglike(c(1, 2))

## [1] -15340915

loglike(c(2, 2))

## [1] -15272717

loglike(c(3, 2))

## [1] -15211448
```

(c) **Write code** to generate a matrix, `z`, of log-likelihood values for 30 θ values between 0.00001 and 0.01 and 30 k values between 0.2 and 2.

[4 marks]

```
## Could use outer(), but requires new log-likelihood function
N <- 30
k <- seq(.2, 2, length.out=N)
theta <- seq(0.00001, .01, length.out=N)
z <- matrix(0, ncol=N, nrow=N)
for (i in seq_along(k)) {
    for (j in seq_along(theta)) {
        z[i, j] <- loglike(c(k[i], theta[j]))
    }
}
```

```
options(width=60, digits=4)
```

The following code and output shows the sequence of k and θ values and shows some features of the `z` matrix.

```
k

##  [1] 0.2000 0.2621 0.3241 0.3862 0.4483 0.5103 0.5724 0.6345
##  [9] 0.6966 0.7586 0.8207 0.8828 0.9448 1.0069 1.0690 1.1310
```

```
## [17] 1.1931 1.2552 1.3172 1.3793 1.4414 1.5034 1.5655 1.6276
## [25] 1.6897 1.7517 1.8138 1.8759 1.9379 2.0000
```

```
theta
```

```
##  [1] 0.0000100 0.0003545 0.0006990 0.0010434 0.0013879
##  [6] 0.0017324 0.0020769 0.0024214 0.0027659 0.0031103
## [11] 0.0034548 0.0037993 0.0041438 0.0044883 0.0048328
## [16] 0.0051772 0.0055217 0.0058662 0.0062107 0.0065552
## [21] 0.0068997 0.0072441 0.0075886 0.0079331 0.0082776
## [26] 0.0086221 0.0089666 0.0093110 0.0096555 0.0100000
```

```
dim(z)
```

```
## [1] 30 30
```

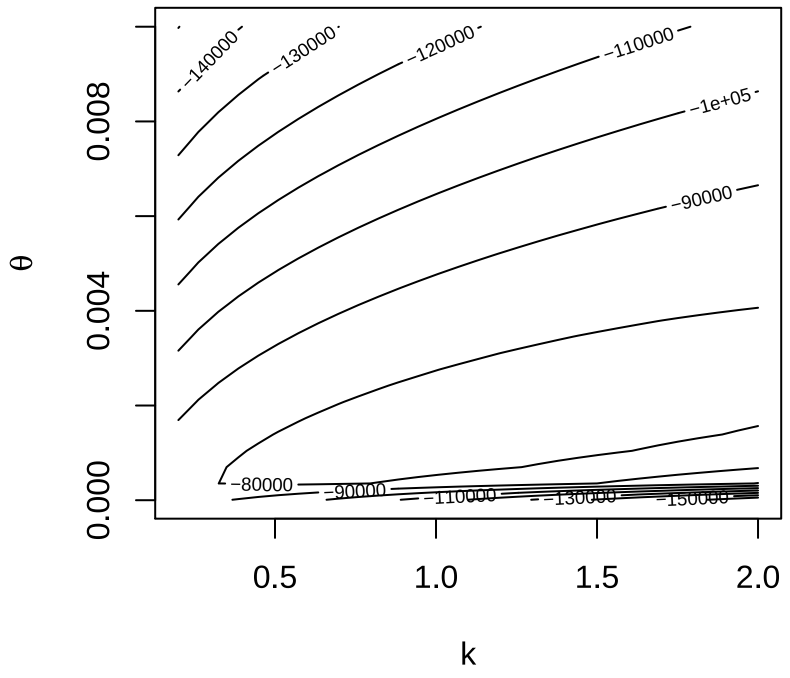
```
z[1:4, 1:4]
```

```
##        [,1]   [,2]   [,3]   [,4]
## [1,] -87343 -82854 -84140 -85982
## [2,] -87826 -81123 -81988 -83581
## [3,] -88932 -80014 -80458 -81803
## [4,] -90453 -79322 -79344 -80441
```

(d) **Write code** to draw a contour plot from the matrix of log-likelihood values like the one shown below.

[2 marks]

```
notrun <- function() {
    par(mfrow=c(2,2), mar=rep(0,4))
    persp(k, theta, zz, ticktype="simple", theta=0, phi=30)
    persp(k, theta, zz, ticktype="simple", theta=-30, phi=30)
    persp(k, theta, zz, ticktype="simple", theta=-60, phi=30)
    persp(k, theta, zz, ticktype="simple", theta=-90, phi=30)
}
contour(k, theta, z, xlab="k", ylab=expression(theta), cex=.7)
```



(e) **Write code** to optimise the log-likelihood function, `loglike`. The result of the optimisation is shown below.

[3 marks]

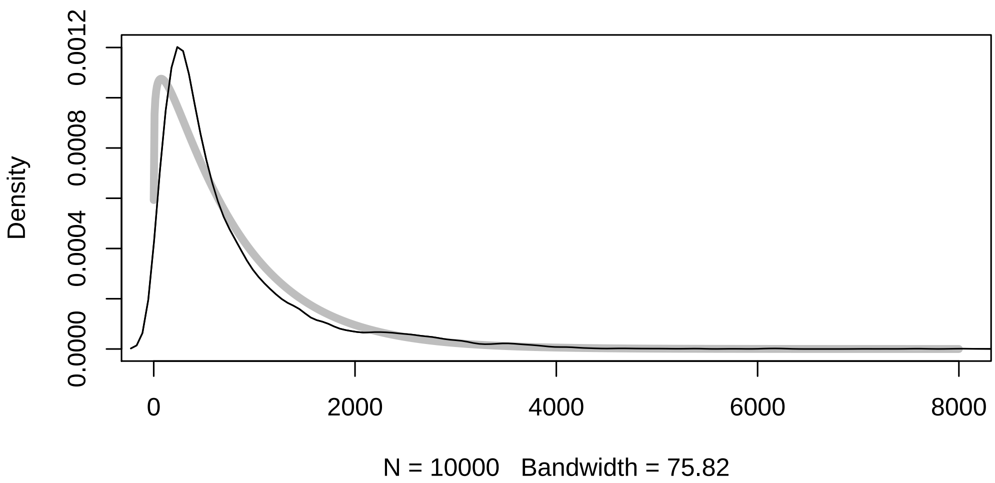
```
optim(c(1, 1), loglike, control=list(fnscale=-1))

## $par
## [1] 1.106297 0.001441
##
## $value
```

```
## [1] -76370
##
## $counts
## function gradient
##      103       NA
##
## $convergence
## [1] 0
##
## $message
## NULL
```

(f) **Write code** that would add a thick grey line to the plot of the distribution of `Trip.Durations` that was shown at the start of this question. The thick grey line should show the gamma distribution that our optimisation has chosen. The plot with the thick grey line added is shown below.

[3 marks]

```
soln <- optim(c(1, 1), loglike, control=list(fnscale=-1))

plot(density(dockless$Trip.Duration), zero.line=FALSE, main="",
     xlim=c(0, 8000))
t <- seq(.1, 8000, length=1001)
lines(t, dgamma(t, soln$par[1], soln$par[2]), col="grey", lwd=5)
lines(density(dockless$Trip.Duration))
```



(g) The following output is from a different optimisation.
Describe the meaning of the values in this result.

[2 marks]

```r
optim(c(1, 1), loglike)
```

```
## $par
## [1] 8.025e+52 6.025e+53
##
## $value
## [1] -4.616e+60
##
## $counts
## function gradient
##      501       NA
##
## $convergence
## [1] 1
##
## $message
## NULL
```

[**20 marks**]