Intro to R Programming: Class Test 2

Class Test Rules & Conditions

- The class test is taken under exam conditions.
- During the test you are not allowed to talk to or otherwise communicate with other students (email, instant messaging, etc.), or access the internet/course material. The only material you can use is the printed copy of R reference manual provided as well as the R help within RStudio.
- **DO NOT** open other web pages - you are only allowed two windows open (R studio and the Moodle Class Test 1 page)
- *Please note we use technological means to check compliance with the above!*
- Students who breech the rules above will be reported to the Clerk of Senate.

Starting the test ...

- You will be already logged in to the computer.
- Log into Moodle and navigate to the Introduction to R Moodle page, Section **2019 Class Test 2.**
- Click on the link **Class Test 2 - MSc** to begin the test.
- The test is password-protected. The password is **markov**.
- You should use an R studio (or R) window to trial and test your answers. Once you have written and tested the code for a question, **copy the code into the corresponding answer field for the question.**
- In the answer field for each question, only include the code with answers for that specific question.

During the test ...

- You can move back and forward through questions during the class test period.
- **If you have any issues logging in to Moodle, or cannot locate the link to start the class test, please let one of the tutors know immediately.**
- **Once open, do not close the moodle browser window.**
- The only external packages you are allowed to use (if you wish, they are not required) are `dplyr` and `ggplot2`.
- If you have experience any issues with the technology throughout the class test, please speak to a tutor immediately.
- For all parts in the test give the R code which can be used to answer the questions. All questions should be answered programatically and should not be hard coded.
- You should only include the code to answer the question, you do not need to include comments or output from the console window.

Important

There are two datasets associated with this test: `ca-fires.Rdata` and `portpirie.Rdata`. You should find both datasets in your computer Desktop. They are also available to download from Moodle, Section **2019 Class Test 2**.

Task 1: California fires

The data frame `ca-fires.Rdata` contains information about fires in the state of California from 1933 until 2016. It has 83 rows and 4 columns. A description of the columns is given below:

Variable	Class	Description
year	numeric	Year
n.of.fires	numeric	Number of fires
acres.burned	numeric	Number of acres burned
dollar.damage	numeric	Cost of damage in US dollars

Table 1: Variables for the `ca-fires.Rdata` data frame.

Set your working directory to the location where your files are stored and enter the line of code below to read the `ca-fires.Rdata` data frame into `R`.

```r
load('ca-fires.Rdata')
```

Q1. [**2 marks**] Identify the year with the highest number of acres burned.

```r
fires$year[which.max(fires$acres.burned)]
```

We want to identify the decade with the highest number of fires. To this end, carry out the following steps:

Q2. [**2 marks**] Create a variable called `dec` which contains a sequence from 1930 to 2020 increasing in units of 10.

```r
dec <- seq(1930, 2020, by=10)
```

Q3. [**3 marks**] Using the sequence `dec` and the `cut` function create a new variable called `decade` which contains for each row the decade that year falls in, i.e. for 1933 the decade will be 1930, for 1990 the decade will be 1990, etc.

```r
decade <- cut(fires$year, dec, right=FALSE, labels=dec[-10])
```

Q4. [**2 marks**] Define a vector `s` which contains the total number of fires per decade.

```r
s <- by(fires$year, decade, sum)
```

Q5. [**2 marks**] Using `s`, find the decade with the highest number of fires. *Note*: If, for example, the highest number of fires occurred in the decade of the 1930s (i.e., in any year from 1930 till 1939), then you should report 1930.

```r
names(which.max(s))
```

! If you have not successfully created the variable `decade` in question 3, then use the following surrogate:

```r
decade = c(rep(1930, 7), rep(1940, 10), rep(1950, 10), rep(1960, 10), rep(1970, 9),
           rep(1980, 10), rep(1990, 10), rep(2000, 10), rep(2010, 7))
```

No marks will be awarded in question 3 if you use the surrogate.

Q6. [**3 marks**] Plot the number of fires versus the cost (dollar damage) in log-scale using a different colour for each decade. Your plot should look like the one below, which was produced using colour i for the i-th decade, $i = 1, 2, \ldots, 9$.

```r
plot(fires$n.of.fires,log(fires$dollar.damage),col=decade,pch=20,xlab = 'Number of fires',
     ylab = 'Cost (log scale)', main = 'Number of fires versus cost (log scale)')
```

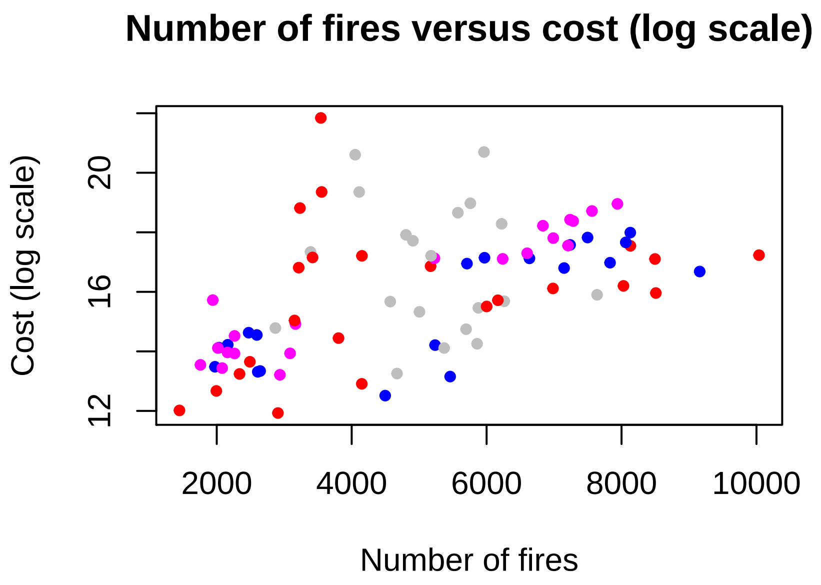

Figure 1: Number of fires versus cost (log scale) using the fires dataset (Q6).

Q7. [**2 marks**] Create a new variable called `year.group`, which takes the following values

$$\texttt{year.group}_i = \begin{cases} 1, & \text{if } \texttt{year}_i < 1974, \\ 2, & \text{if } \texttt{year}_i \geq 1974, \end{cases} \qquad \text{for } i = 1, \ldots, 83.$$

! If you have not successfully created the variable `year.group`, then use the following surrogate:

```r
year.group <- c(rep(1,41), rep(2,42))
```

No marks will be awarded for Question 7 if you use the surrogate.

```r
year.group <- ifelse(fires$year<1974, 1, 2)
```

We want to test the hypothesis that the two groups have equal means of acres burned. To this end, carry out the following steps:

Q8. [**3 marks**] Compute t given below (t is known as the Welch's t-test).

$$t = \frac{\overline{x}_1 - \overline{x}_2}{\Delta_{12}}, \qquad \text{where} \quad \overline{x}_1 = \frac{1}{n_1}\sum_{j=1}^{n_1} x_{1j}, \quad \overline{x}_2 = \frac{1}{n_2}\sum_{j=1}^{n_2} x_{2j},$$

and

$$s_1^2 = \frac{1}{n_1 - 1}\sum_{j=1}^{n_1}(x_{1j} - \overline{x}_1)^2, \quad s_2^2 = \frac{1}{n_2 - 1}\sum_{j=1}^{n_2}(x_{2j} - \overline{x}_2)^2, \quad \Delta_{12} = \sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}$$

Note that x_{ij} corresponds to the j-th acre burned when `year.group == i`, $i = 1, 2$. Use the built-in functions `mean` and `var` to compute $\overline{x}_i$ and s_i^2, respectively.

Hint: you can check your result using the built-in function `t.test(x1,x2)`.

```r
x1 = fires$acres.burned[year.group == 1]
x2 = fires$acres.burned[year.group == 2]
```

```r
n1 = length(x1)
n2 = length(x2)
s1.2 = var(x1)
s2.2 = var(x2)
delta = sqrt(s1.2/n1 + s2.2/n2)
t = (mean(x1) - mean(x2))/delta; t
t.test(x1, x2)$statistic
```

Q9. [**2 marks**] The test rejects the null hypothesis of equal means if $t > t^\star = 1.99$. Define a variable called `out.t.test` which is `TRUE` if $t > t^\star$ and `FALSE` otherwise.

```r
out.t.test = t > 1.99; out.t.test
```

Task 2: The generalised extreme-value distribution

The generalised extreme-value (GEV) distribution is a family of continuous probability distributions developed within extreme value theory to model maximum values. The GEV can be parametrised in terms of a location parameter $\mu \in \mathbb{R}$, a scale parameter $\sigma > 0$, and a shape parameter $\xi \in \mathbb{R}$, and its distribution function is given by

$$G(z) = \exp\left\{-\left[1 + \xi \cdot \left(\frac{z-\mu}{\sigma}\right)\right]^{-1/\xi}\right\}, \qquad \text{defined in} \quad \mathcal{Z} = \left\{z : 1 + \xi \cdot \left(\frac{z-\mu}{\sigma}\right) > 0\right\}$$

Q10. [**4 marks**] Create a function called `pgev` that receives four arguments: z, μ, σ, and ξ. Your function should return $G(z)$ when $\sigma > 0$ and $z \in \mathcal{Z}$. If $z \notin \mathcal{Z}$, i.e., if z is such that $1 + \xi \cdot \left(\frac{z-\mu}{\sigma}\right) \leq 0$, then your function should return 0. If $\sigma \leq 0$, then your function should return a warning message saying `invalid scale`.

```r
pgev = function(z, mu, sigma, xi){
  z. = 1+xi*(z-mu)/sigma

  if ( sigma >0 & z.>0) {
    return(exp(-z.^(-1/xi)))
  } else if ( z.<=0) {
    return(0)
  } else {
    return(warning('invalid scale'))
  }
}
```

Q11. [**3 marks**] The `portpirie.Rdata` data frame has 65 rows and 2 columns. The second column gives annual maximum sea levels recorded at Port Pirie, South Australia, from 1923 to 1987. The first column gives the corresponding years. Set your working directory to the location where your files are stored and run the line of code below to read the `portpirie.Rdata` data frame into `R`.

```r
load('portpirie.Rdata')
```

Use `sapply` to evaluate your function `pgev` over the annual maximimum sea levels recorded at Port Pirie, with $\mu = 3.87$, $\sigma = 0.198$, and $\xi = -0.05$. Save your results in a vector called `pgev.portpirie`.

```r
pgev.portpirie = sapply(portpirie$SeaLevel, pgev, mu = 3.87, sigma = 0.198, xi = -0.05)
```

Task 3: Generating samples from a Gamma distribution

You have seen in your Probability courses that the sum of independent exponential distributions with the same rate θ has a gamma distribution, i.e., if $X_1, \ldots, X_k \overset{\text{i.i.d.}}{\sim} \mathsf{Exp}(\theta)$, then $Y = \sum_{i=1}^{k} X_i \sim \mathsf{Ga}(k, \theta)$. We can exploit this to draw a sample from the $\mathsf{Ga}(k, \theta)$ distribution for $k \in \mathbb{N}$. Specifically, to draw one realisation from the $\mathsf{Ga}(k, \theta)$ we can draw k exponentially distributed random numbers (with rate θ) and compute their sum. The sum then has the desired $\mathsf{Ga}(k, \theta)$ distribution.

Q12. [**4 marks**] Using the argument above, write a function called `sample.gamma` which take n, k, and θ as arguments and which returns a sample of size n from the $\mathsf{Ga}(k, \theta)$ distribution. No marks will be awarded for this question if the `rgamma` function is used here.

Hint: You can draw m values from the $\mathsf{Exp}(\theta)$ distribution by calling the function `rexp(m, theta)`.

```
sample.gamma <- function(n, k, theta){
  out = rep(NA, n)
  for(i in 1:n)
    out[i] = sum(rexp(k, theta))
  out
}
```

Q13. [**2 marks**] Run the line of code below

```
set.seed(123)
```

Then use your function `sample.gamma` to draw a sample of size $n = 100$ from the $\mathsf{Ga}(5, 2)$ distribution.

```
sample.gamma(100, 5, 2)
```

The method described above to draw samples from the gamma distribution only works if the shape parameter k is a natural number ($k \in \mathbb{N}$). To draw samples from the $\mathsf{Ga}(\alpha, \theta)$ distribution for $\alpha \notin \mathbb{N}$, we need to use a different strategy. We will only cover the simpler case that both $\theta > 1$ and $\alpha > 1$. In that case, we can sample from the $\mathsf{Ga}(\alpha, \theta)$ distribution using a method called **rejection sampling**.

The basic idea of rejection sampling is that rather than sampling from the target distribution, which is difficult, we sample from a different distribution, which is easy to sample from (in our case this is the gamma distribution with natural shape parameter). However, given that we have then drawn samples from the "wrong" distribution, we have to adaptively reject some of the values we have generated. This correction makes sure we are sampling from the correct target distribution.

To generate one value Y from the $Ga(\alpha, \theta)$ distribution we can carry out the following steps:

i) Using your function `sample.gamma`, draw a value $X \sim Ga(k, \theta - 1)$, where $k = \lfloor \alpha \rfloor$ can be computed in R using `floor(alpha)`.

ii) Draw $U \sim \mathcal{U}(0, 1)$.

iii) If $U \leq p$ set $Y = X$. Otherwise, return to i).

The probability p is given by

$$p = \frac{f_{(\alpha,\theta)}(X)}{M \cdot f_{(k,\theta-1)}(X)}, \quad \text{where} \quad M = \frac{f_{(\alpha,\theta)}(\alpha - k)}{f_{(k,\theta-1)}(\alpha - k)},$$

and $f_{(\alpha,\theta)}(x)$ is the probability density function of the gamma distribution, which can be found in R using `dgamma(x, shape=alpha, rate=theta)`.

Q14. [**6 marks**] Using rejection sampling, write a function called `sampleGammaRej` which takes the desired sample size n as well as the parameters α and θ as arguments and which returns a sample of size n from the $\mathsf{Ga}(\alpha, \theta)$ distribution. Your function should stop and return the message `invalid parameters` if $\theta \leq 1$ or $\alpha \leq 1$.

```
sampleGammaRej <- function(n, alpha, theta){
  if(theta <= 1 | alpha <= 1) stop('invalid parameters')
  else{
    iter = 0
    y = NULL
    repeat{
      iter = iter + 1
      k = floor(alpha)
      x = sample.gamma(1,k,theta)
      u = runif(1)
      M = dgamma(alpha-k, shape = alpha, rate = theta)/dgamma(alpha-k, shape = alpha,
                                                         rate = (theta-1))
      p = dgamma(x, shape = alpha, rate = theta)/(M * dgamma(x, shape = alpha,
                                                        rate = (theta-1)))

      if(u <= p)
        y = c(y,x)
      if(iter == n) break
    }
    return(y)
  }
}
```