STAT 210
Applied Statistics and Data Analysis:
Homework 1 - Solution

Due on Sept. 11/2022

Question 1

Consider the following system of equations:

$$
\begin{aligned}
3x + 2y + 2z + 4w &= 28 \\
2x + y + z &= 14 \\
2x + 5z + 5w &= 28 \\
6x + 2y + 2z + w &= 37
\end{aligned}
$$

1. Create a matrix in `R` with the coefficients of the system, and a vector with the constants on the right-hand side of the equations. Call them `mat1` and `vec1`, respectively.

```
(mat1 <- matrix(c(3,2,2,4,2,1,1,0,2,0,5,5,6,2,2,1),
                byrow = T, ncol = 4))
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    3    2    2    4
## [2,]    2    1    1    0
## [3,]    2    0    5    5
## [4,]    6    2    2    1
```

```
(vec1 <- c(28, 14, 28, 37))
```

```
## [1] 28 14 28 37
```

2. Find the inverse of `mat1` and call it `mat2`.

To find the inverse of a matrix we use the function `solve`:

```
(mat2 <- solve(mat1))
```

```
##            [,1]       [,2] [,3]       [,4]
## [1,] -0.1111111 -0.6666667  0.0  0.4444444
## [2,]  0.4000000  1.4000000 -0.2 -0.6000000
## [3,] -0.1777778  0.9333333  0.2 -0.2888889
## [4,]  0.2222222 -0.6666667  0.0  0.1111111
```

We can verify that this is the inverse by multiplying it by `mat1`. We round off the result to 15 decimals.

```
round(mat1 %*% mat2, 15)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
## [4,]    0    0    0    1
```

3. Create a list named `list1` having as components `mat1`, `vec1`, and `mat2`. Call these components `item1`, `item2`, and `item3`, respectively.

We use the command `list` to create a list:

```
list1 <- list(item1 = mat1, item2 = vec1, item3 = mat2)
```

4. Remove `mat1`, `vec1`, and `mat2` from the working directory.

We use the function `rm` to clear objects from the working directory.

```
rm(mat1, vec1, mat2)
```

5. Solve the system of equations and call the solution `vec2`.

We use again the function `solve`, but with two arguments now.

```
(vec2 <- solve(list1$item1,list1$item2))
```

```
## [1] 4 3 3 1
```

6. Verify the solution.

```
list1$item1%*%vec2
```

```
##      [,1]
## [1,]   28
## [2,]   14
## [3,]   28
## [4,]   37
```

7. Verify that if you multiply the inverse matrix `mat2` by `vec1` you get the solution.

```
list1$item3%*%list1$item2
```

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

8. Find the eigenvalues of `mat1` and `mat2` and verify that the eigenvalues of `mat2` are the reciprocals of the eigenvalues of `mat1`.

   The eigenvectors and eigenvalues are obtained with the function `eigen`. We find and store the eigenvalues and eigenvectors for both matrices.

```
eigen1 <- eigen(list1$item1)
eigen2 <- eigen(list1$item3)
```

To extract the eigenvalues we look at the structure of the object:

```
str(eigen1)
```

```
## List of 2
##  $ values : num [1:4] 10.334 -3 1.903 0.763
##  $ vectors: num [1:4, 1:4] -0.51 -0.181 -0.669 -0.51 -0.487 ...
##  - attr(*, "class")= chr "eigen"
```

We print the eigenvalues for `mat1` and the reciprocals of the eigenvalues for `mat2`:

```
eigen1$values
```

```
## [1] 10.3342514 -3.0000000  1.9030233  0.7627252
```

```r
1/eigen2$values
```

```
## [1]  0.7627252  1.9030233 -3.0000000 10.3342514
```

and the values are the same.

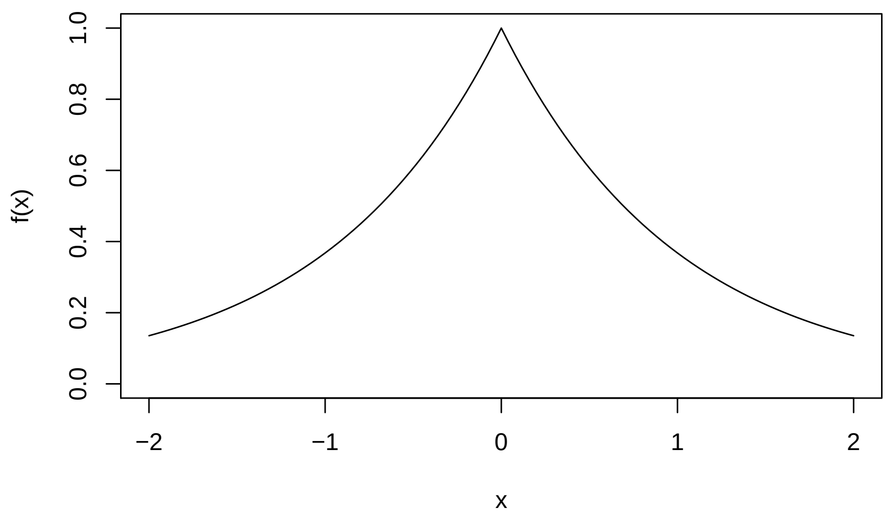
Question 2

Consider the function $f(x) = e^{-|x|}$, for $x \in \mathbb{R}$. We want to use the MonteCarlo method to estimate the value of the integral

$$\int_{-2}^{2} f(x)\,dx$$

1. Plot a graph of this function in the region where you want to calculate the integral.

```r
curve(exp(-abs(x)),-2,2, ylab = 'f(x)', ylim = c(0,1))
```



2. Generate $N = 1000$ random numbers with uniform distribution in the rectangle $[-2, 2] \times [0, 1]$. Count how many points fall below the curve $f(x) = e^{-|x|}$ and estimate the integral using the fraction of these points with respect to the total number of points and the area of the rectangle. Call the estimator I_{1000}

```r
set.seed(4567)
x <- runif(1000,-2,2)
y <- runif(1000,0,1)
(points.below <- sum(y <= exp(-abs(x))))
```

```
## [1] 466
```

```r
(I1000 <- 4*points.below/1000)
```

```
## [1] 1.864
```

We have that $I_{1000} = 1.864$.

3 Compute analytically the value of the integral and compare with the approximation you obtained in 3. Call I the value of the integral and calculate $|I - \bar{I}_{1000}|$

$$I = \int_{-2}^{2} e^{-|x|}\,dx = 2\int_{0}^{2} e^{-x}\,dx = 2\big(-e^{-x}\big|_0^2 = 2(1 - e^{-2}\big)$$

and we can calculate the value of this expression using R:

```r
(I = 2*(1-exp(-2)))
```

```
## [1] 1.729329
```

The error is $|I - \bar{I}_{1000}| = 0.1346706$.

4 Repeat for $N = 10^k$ for $k = 4, 5, \ldots, 8$ and compute the deviation $|I - \bar{I}_N|$ from the exact result.

```r
error <- numeric(6)
for (n in 3:8) {
  N <- 10^n
  x <- runif(N,-2,2)
  y <- runif(N,0,1)

  (points.below <- sum(y <= exp(-abs(x))))
  (Area1 <- 4*points.below/N)
  error[n-2]  <- abs(2*(1-exp(-2))-Area1)
  }
error
```

```
## [1] 0.0613294335 0.0021294335 0.0024894335 0.0013025665 0.0001917665
## [6] 0.0002558335
```

5 Do a log-log plot of the deviation as a function of N. The points should follow approximately a straight line.

```r
plot(3:8,log(error))
```