STOCKHOLM UNIVERSITY

Department of Statistics

Parfait Munezero

R PROGRAMMING EXAM

December 5, 2023

Time : 8:00 - 13:00

Results will be announced no later than December 20, 2023.

The exam should be submitted no later than 13.00 (1PM), please carefully read the provided instructions to digital exams in Safe Exam Browser (SEB).

The exam is to be written as a R Markdown report. You need to submit the R markdown file (`.Rmd`) as well as an output file either in `.pdf` or `.html` format.

Task 1: Plotting with ggplot (10 points)

In this task you are going to use `ggplot2` to create some basic plots. You will use the data `mpg` that is available in `ggplot` package.

1. Load the data `mpg` and print the first 10 observations. **(2 points)**
2. Use the `geom_point()` to create a scatter plot of the city mileage (cty) vs highway mileage (hwy). **(2 points)**
3. Now use the `geom_jitter()` to create the same scatter plot as in Q.2, and comment on the difference. **(2 points)**
4. Modify your plot in Q.3 to highlight the effect of the drive mode (drv) and number of cylinders (cyl) by setting the color and size of the points depending on the value of `drv` and `cyl respectively`. Add appropriate axes' label and title. **(4 points)**

Tast 2: Random number generation (15 points)

In this task you are going to simulate random samples from a bivariate normal distribution. There are various ways to do so in R, here you will use the property that:

> If X and Y are jointly normal random variables, then the conditional distribution of X given Y is a normal distribution with mean
>
> $$E(X|Y) = E[X] + \rho\frac{\sigma_x}{\sigma_y}(Y - E[Y])$$

and variance

$$\sigma^2(X|Y) = (1-\rho^2)\sigma_X^2$$

where σ_x and σ_y are standard deviations of X and Y respectively, and ρ is the correlation between X and Y. The conditional mean and variance $E(Y|X)$ and $\sigma^2(Y|X)$ of Y given X can also be computed following the equations above as: $E(Y|X) = E[Y] + \rho\frac{\sigma_y}{\sigma_x}(X - E[X])$ and $\sigma^2(Y|X) = (1-\rho^2)\sigma_Y^2$.

One way of sampling from the joint distribution of X and Y is to draw samples from the conditional normal distributions $X|Y$ and $Y|X$ interchangeably for a given number of iterations N; see the algorithm below.

Algorithm 1: Sampling from Bivariate Normal distribution

Inputs: N, $E[X]$, $E[Y]$, ρ, σ_x, σ_y

---

1. Initialize X and Y to zero.
2. Create an empty matrix **mat_xy** with N rows and 2 columns.
3. for each iteration **i** in [1:N] do:
   a. Compute $E[X|Y]$ and $\sigma^2(X|Y)$ conditioned to the most recent value of Y
   b. Sample X: draw one random number X from the normal distribution with mean and variance computed in (a).
   c. Compute $E[Y|X]$ and $\sigma^2(Y|X)$ conditioned to the most recent value of X.
   d. Sample Y: draw one random number Y from the normal distribution with mean and variance computed in (c).
   e. Save X and Y into the matrix **Mat_xy** on the ith row
4. Return the matrix **mat_xy**.

---

1. Write your own function to implement each step in **Algorithm 1**. Your function should require $N, E[X], E[Y], \rho, \sigma_x, \sigma_y$ as arguments. Name the function to **biv_rnorm**. **(5 points)**

2. Use your function **biv_rnorm** to simulate a sample $N = 1000$ random numbers from the bivariate normal distribution with $E[X] = 0, E[Y] = 0, \rho = -0.7, \sigma_x = 1, \sigma_y = 2$. **(2 point)**

3. Plot a scatter plot of the sampled X and Y: Add axes labels and title. **(2 points)**

4. Now use the standard `rmvnorm()` from the library `mvtnorm` to sample from the bivariate normal distribution with the vector mean [0,0] and covariance matrix. **(2 points)**

5. Plot the scatter plot of X and Y for both the sample in Q.2 and in Q.4 in one plot. **Hint:** plot one sample and then use the function `points()` to add the other sample. Differentiate the two samples by color. **(4 points)**

Task 3: Vector manipulation (10 points)

Consider the following two vectors `x` and `y` of length 250.

```
set.seed(50)
x <- sample(0:999, 250, replace=T)
y <- sample(0:999, 250, replace=T)
```

1. Compute the vector $yx\_diff = (y_2 - x_1, y_3 - x_2, \dots, y_n - x_{n-1})$, where $n = 250$. **(4 points)**

2. Print

   (a) The index positions in `yx_diff` of the values that are > 800. **(1 point)**

   (b) The values in `yx_diff` that are > 800. **(1 point)**

3. Calculate $\sum_{i=1}^{n-1} \left(\frac{sin(y_i)}{cos(x_{i+1})}\right)$. **(4 points)**

Task 4: Simple functions (15 points)

Consider the function:

$$f(x) = \begin{cases} x^2 + 2x + 3 & if \;\; x < 0 \\ x + 3 & if \;\; 0 \leq x < 2 \\ x^2 + 4x - 7 & x \geq 0 \end{cases}$$

1. Write a function `f_x` which takes a single argument `x` and return the value of $f(x)$. **(5 points)**

2. Plot the curve of $f(x)$ for $-3 \leq x \leq 3$. To do this, Create a sequence of x values between -3 and 3 with a step size of 0.5 and compute $f(x)$ for each x. **(3 points)**

3. Write the function `recurse_fn` which takes a single argument n and returns all values computed from the following recursive equation:

$$x_j = x_{j-2} + \frac{2}{x_{j-1}}$$

   for $j = 1,2, \ldots, n$. Use the starting values $x_0 = 0$ and $x_1 = 1$. **(5 points)**

4. Run the function above with $n = 200$ and plot the resulting series. Use the function `plot.ts()`. **(2 points)**

Good luck!