Problem Set 4

In this problem set, you will continue to be asked to work with the gamma distribution and you may find it useful to refer back to the previous problem set and/or solutions. I recommend Wikipedia as a reference, `https://en.wikipedia.org/wiki/Gamma_distribution`.

Keep your rendered `.pdf` to no more than 4 pages. Only provide code in the rendered `.pdf` when it is specifically requested.

1. This question refers back to 2. on the previous homework assignment. Create a matrix with 9 rows and 10,000 columns called `Y` with elements given by `sample((-3):3, 9*10000, replace = TRUE)`.

(a) Repeat 2. (c)-(e) from the previous problem set using `apply`. Print the code you use to create these matrices and vectors to the rendered `.pdf`.

(b) Repeat 2. (f) from the previous problem set with one addition - include points that correspond to the values in `x` on the x-axis and 0 on the y-axis that use a point type that corresponds to a $\times$ symbol on the first plot. You may find the information provided by typing `help(pch)` in your console helpful. Also, add an additional point to the legend that refers to your $\times$ symbols and annotate it with "Truth" for the plot that contains the $\times$ symbols.

(c) In at most one sentence, describe why the plot you made in the previous part and the plot you made for 2. (f) on Problem Set 3 differ and which you prefer. Use what you know about statistics.

(d) Now, create a matrix with 9 rows and 10,000 columns called `Y` with elements given by `x + sample((-3):3, 9*10000, replace = TRUE)`. Repeat (c) for this matrix, this time including points that correspond to the values in `x` on the x-axis and `x` again on the y-axis and use a point type that corresponds to a $\times$ symbol on the first plot. Again add an additional point to the legend that refers to your $\times$ symbols and annotate it with "Truth" for the plot that contains the $\times$ symbols.

2. Continue using four sets of values for the shape parameter α and rate parameter λ: $(\alpha_1, \lambda_1) = (0.5, 0.5)$, $(\alpha_2, \lambda_2) = (2, 1)$, $(\alpha_3, \lambda_3) = (0.5, 1)$, and $(\alpha_4, \lambda_4) = (2, 0.5)$, and store these values in a matrix `Y` with 4 rows and 2 columns, where the first column corresponds to the shape parameter, the second column corresponds to the rate parameter, and the rows correspond to the parameters in the order provided above.

(a) Create a matrix `W` that with 4 rows and $100,000$ columns. Fill each row of `W` with $100,000$ draws from the corresponding gamma distribution using the `rgamma` function. Print the code you use to create this matrix to the rendered `.pdf`.

(b) Create a matrix `V` that with 4 rows and $100,000$ columns. Fill each row of `V` with $100,000$ draws from the corresponding gamma distribution using the `rgamma` function obtained by simulating from gamma distributions with shape 0.5 or 2 and rate 1 using `rgamma` and applying simple mathematical operations. Print the code you use to create this matrix to the rendered `.pdf`. You may need to remind yourself of some gamma distribution facts.

(c) Again, consider a sequence of values given by `x <- seq(0.01, 6, by = 0.01)`. Create two matrices `pW` and `pV` each with the same number of rows as `W` and the same number of columns as the length of `x` minus 1 with all elements equal to `NA`. Fill each element of `pW` and `pV` with the proportion of elements in the corresponding row of `W` or `V` between the corresponding element of `x` and the next largest element of `x`. Print the code you use to create these matrices matrix to the rendered `.pdf`.

(d) Create a vector `xmid` that has the same number of elements as the length of `x` minus one, and set the elements of `xmid` to be the midpoints between each element of `x`. Print the code you use to create this matrix to the rendered `.pdf`.

(e) Create a single plot that includes 8 lines corresponding to each row of `pW` and each row of `pV` plotted against `xmid`. Use different colors lines to reflect the parameters of the gamma distribution each row of `pV` or `pW` describes and use different line types for whether or not the line was computed from `pV` or `pW`. Make sure that your plot is clearly annotated and has an informative legend.

(f) Based on what you find in (e), explain in at most one sentence whether or not the approaches for simulating gamma random variables in (a) and (b) are equivalent.

(g) Create two matrices `qW` and `qV` each with the same number of rows as `W` and the same number of columns as the length of `x` with all elements equal to `NA`. Fill each element of `qW` and `qV` with the proportion of elements in the corresponding row of `W` or `V` that are less than or equal to the corresponding element of `x`. Create a third matrix `qT` with the same dimensions as `qW` and `qV`. Fill each element of `qT` with the probability that a gamma random variable with parameters corresponding to the same row of `Y` is less than or equal to the corresponding element of `x`. Use one of the functions that starts with a letter and ends with `gamma`. Print the code you use to create `qT` to the rendered `.pdf`.

(h) Create a single plot that includes 12 lines corresponding to each row of `qW`, each row of `qV`, and each row of `qT` plotted against `x`. Use different colors lines to reflect the parameters of the gamma distribution each row of `qV`, `qW`, or `qT` describes and use different line types for whether or not the line was computed from `qV`, `qW`, or `qT`. Make sure that your plot is clearly annotated and has an informative legend.

(i) In at most two sentences, explain the differences between lines corresponding to the same gamma distribution in (h), if there are any.

3. This question is introducing the multivariate normal distribution, which I don't expect you to be familiar with yet, and the computational challenges that come up when we work with it.

(a) Simulate 5 independent standard normal variables and store them in a vector called `z`. Print the code you use to create `z` to the rendered `.pdf`.

(b) Write a for loop that generates 5 normal random variables stored in a vector `y` using `z` and basic mathematical operations, where the first element of `y` has a standard normal distribution and the remaining elements of `y` conditional on previous elements have a normal distribution with a mean that of 0.5 times the last element of `y` and variance 0.75. Print the code you use to create `z` to the rendered `.pdf`.

(c) Create a 3×3 matrix D that creates the same vector `y` from `z` if you apply `D%*%z`. You'll need to do some algebra. Print the code you use to create `z` to the rendered `.pdf`.

(d) Print one line of code you could use to check that your construction of `D` in (c) is correct based on what you obtained in (b).

(e) Using the `arcovariance.R` code, create a 3×3 version of the matrix `C`. You could just take the first 3 rows and first 3 columns of `C`. Print the code you need to compute the Cholesky decomposition of this 3×3 matrix and take its transpose the rendered `.pdf`.

(f) Create a plot of the nonzero elements of D against the nonzero elements of the transposed Cholesky decomposition in (e). Make sure that your plot is clearly annotated.