CPMA 573 — Homework #3

Before you begin: precede your R script with `pdf(file='myfile.pdf')`, and include as the last line `dev.off()`. This will funnel every plot you make into a single pdf to submit along with your code. While you are working in Rstudio, I recommend commenting out these two lines so that you can see the graphs as you go. When you are done, uncomment them and re-run your whole script to generate the file with the figures. Make sure all of your figures have informative titles (using the option `main='mytitle'` within most plotting commands e.g. `hist()` and `curve()`).

Q1 : Recall the beta density function $f(x|\alpha,\beta)$ given by

$$f(x|\alpha,\beta) \;=\; \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1},$$

for $x \in (0,1)$ and positive α and β. Let $X \sim \text{beta}(2,5)$. Find $P(X > \frac{1}{2})$ by:

a) Monte Carlo (MC) integration.

b) Simulating $25{,}000$ beta$(2,5)$ realizations. Use rejection sampling and uniform proposal density $q(x)$.

c) Using the following command to find the theoretical solution: `pbeta(0.5, shape1=2, shape2=5, lower.tail=FALSE)`. Note: without the `lower.tail=FALSE` argument, this would return $P(X < \frac{1}{2})$

Q2 : Let X and Y be independent standard normal random variables. Use the polar method to simulate $25{,}000$ pairs (x,y) of independent, standard normal realizations.

a) Use these realizations to estimate $P(X > Y)$

b) Use these realizations to find the value of k that satisfies $P(\sqrt{X^2+Y^2} < k) = \frac{1}{2}$

c) Estimate $P(-1.96 < X < 1.96)$ for $X \sim \text{Normal}(0,1)$ using:

   i. your X draws

   ii. MC integration

   iii. the command `pnorm(1.96)-pnorm(-1.96)` (this will give you the theoretical answer!)

Q3 : Let X and Y have jointly normal density $f(x,y)$ given by:

$$f(x,y) \;=\; \frac{1}{2\pi\sqrt{1-\rho^2}} \cdot \exp\left\{-(x^2+y^2-2\rho xy)/(2(1-\rho^2))\right\} \;.$$

Write a function that uses Gibbs sampling to produce realizations from this joint density, where $\rho = 0.75$. Your program should be flexible enough to accommodate a choice of

- Correlation ρ
- Initial values x_0.
- Number of desired (x_i, y_i) realizations.
- Number of iterations to skip (lag) in order to avoid serial correlation (a lag of 1 should be the same as no lag)
- Number of burn-in iterations to skip

Note that the number of desired realizations, together with the burn-in and lag, should determine the number of required iterations. Use this program to complete the following:

a) Generate 500 realizations with $x_0 = 80$, no burn-in and no lag. Plot these realizations and connect the points in order using the following commands:

```
plot(xvec[1:500],yvec[1:500])
arrows(xvec[1:499],yvec[1:499],xvec[2:500],yvec[2:500])
```

b) Use `ts.plot()` to draw time series plots for both x and y. Use these two plots to choose a reasonable burn-in.

c) Use `acf()` to look at autocorrelation plots for both your x and y draws. Use these two plots to choose a reasonable lag for generating independent realizations of (x_i, y_i).

d) Using the lag and burn-in from parts b) and c) draw 10,000 independent realizations of X. Plot a histogram of x, and overlay the theoretical standard normal curve to confirm that the marginal density is correct.

Q4 : Use your Gibbs sampling function from above to generate 15,000 independent realizations each for $\rho = \{-0.9, -0.8, ..., 0.8, 0.9\}$ (note: the function `seq` may be useful here). For each value of ρ, approximate $P(X > 0 \text{ and } Y > 0)$, and plot these probabilities as a function of ρ, setting your vertical axis limits to (0,1) using `ylim`.