Stat 580 (Spring 2017): Homework 3

Due, in class, March 2, 2017

Please submit your homework with codes (hard copy) in class and upload the corresponding codes to the Blackboard. Problems marked with * will be graded in detail and they are worth 50% of the total score. Remaining problems, worth the remaining 50% of the total score, will be given full mark if reasonable amount of work is shown.

1. Prove the two equalities in the slide set 5 page 18:

   (a) $P\{U \le r(X)\} = \frac{1}{\alpha}\int_{\mathcal{X}} q(x)dx$

   (b) $P\{X \in A, U \le r(X)\} = \frac{1}{\alpha}\int_{A} q(x)dx$

   And show that the rejection sampling algorithm generates the desired random variable. That means, show $P(Y \in A) = \int_A f(x)dx$. (Notations follow the slide set.)

2. * Consider the following two density functions:

$$f(x) \propto \sqrt{4+x}x^{\theta-1}e^{-x}, \quad g(x) \propto (2x^{\theta-1} + x^{\theta-1/2})e^{-x}, \quad x > 0.$$

   (Assume that you can simulate from Gamma distribution directly.)

   (a) Find the value of the normalizing constant for $g(x)$.

   (b) Show that $g(x)$ is a mixture of Gamma distributions. (i.e., $g(x)$ can be written in the form $\sum_{j=1}^{J} w_j g_j(x)$, where $w_j > 0$, $\sum_{j=1}^{J} w_j = 1$, and g_j is the density function of a Gamma distribution. Here each g_j corresponds to a component.) Identify the component distributions and their weights in the mixture.

   (c) Design a procedure to sample from $g(x)$.

   (d) Design a procedure using rejection sampling to sample from $f(x)$ using $g(x)$ as the proposal distribution.

3. Using LAPACK, write a C program that performs a linear regression (with intercept) and returns the regression cofficients. For consistency, use the following symbolic constants:

```
#define N 16  /* number of observations */
#define P 2   /* number of predictors */
```

   And the data should be defined in the program. For example:

```
/* longley dataset from R: Employed (Y) GNP.deflator and Population (X) */
double Y[N] = {60.323,61.122,60.171,61.187,63.221,63.639,64.989,
               63.761,66.019,67.857,68.169,66.513,68.655,69.564,
               69.331,70.551};
double X[N][P] =
{{83,107.608},
{88.5,108.632},
{88.2,109.773},
{89.5,110.929},
{96.2,112.075},
{98.1,113.27},
{99,115.094},
{100,116.219},
{101.2,117.388},
{104.6,118.734},
{108.4,120.445},
{110.8,121.95},
{112.6,123.366},
{114.2,125.368},
{115.7,127.852},
{116.9,130.081}};
```

The corresponding output for this dataset should be:

```
The regression coefficients: 26.851352 0.240842 0.119026
```

Note that your program should perform the corresponding linear regression if one replaces the above symbolic constants (N and P) and data variables (X and Y) correctly.

4. * Given a data matrix $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_P) \in \mathbb{R}^{N \times P}$ (N observations, P variables, with $P < N$), write a program (using LAPACK) to compute the principal component scores for observations in $\mathbf{X}$ described as follows. Let $\mathbf{z}_j = \mathbf{x}_j - (\mathbf{x}_j^\intercal \mathbf{1}/N)\mathbf{1}$ for $j = 1, \ldots, P$, where $\mathbf{1} = (1, \ldots, 1)^\intercal$ is a vector of length N. Note that $\mathbf{x}_j^\intercal \mathbf{1}/N$ is the mean of elements in $\mathbf{x}_j$ and therefore $\mathbf{z}_j$ is simply the centered version of $\mathbf{x}_j$. Write $\mathbf{Z} = (\mathbf{z}_1, \ldots, \mathbf{z}_P)$. Define $\mathbf{I}_P$ as the P-by-P identity matrix. Then the principal component analysis can be conducted through the singular value decomposition of $\mathbf{Z}$:

$$\mathbf{Z} = \mathbf{U}\mathbf{D}\mathbf{V}^\intercal,$$

where $\mathbf{U}$ is an $N \times P$ orthogonal matrix (i.e., $\mathbf{U}^\intercal\mathbf{U} = \mathbf{I}_P$), and $\mathbf{V}$ is a $P \times P$ orthogonal matrix (i.e., $\mathbf{V}^\intercal\mathbf{V} = \mathbf{I}_P$), and $\mathbf{D}$ is a $P \times P$ diagonal matrix with diagonal elements $d_1 \geq d_2 \geq \cdots \geq d_P \geq 0$. The principal component scores are given by $\mathbf{UD}$. For example, the principal component scores for the first observation are given by the first row of $\mathbf{UD}$.

For consistency, use the following symbolic constants:

```
#define N 16  /* number of observations */
#define P 2   /* number of predictors */
```

And the data should be defined in the program. For example:

```
/* longley dataset from R */
double X[N][P] =
{{83,107.608},
{88.5,108.632},
{88.2,109.773},
{89.5,110.929},
{96.2,112.075},
{98.1,113.27},
{99,115.094},
{100,116.219},
{101.2,117.388},
{104.6,118.734},
{108.4,120.445},
{110.8,121.95},
{112.6,123.366},
{114.2,125.368},
{115.7,127.852},
{116.9,130.081}};
```

The corresponding output for this dataset should be:

```
The principal component scores:
21.027360 1.786917
15.841324 -0.311559
15.479775 0.811456
13.761879 1.085622
7.498950 -1.556165
5.254453 -1.572194
```

```
3.513952 -0.519752
2.065548 -0.110180
0.424918 0.228784
-3.164887 -0.467648
-7.288287 -1.071878
-10.121032 -1.095894
-12.400256 -0.871850
-14.826466 -0.046314
-17.427933 1.239231
-19.639299 2.471424
```

(Note that the signs of the scores are not unique.)

5. Write a C program that (a) sorts an array via insertation sort algorithm:
   http://en.wikipedia.org/wiki/Insertion_sort
   and (b) prints the median. The data should be defined in the program with symbolic constant N representing the number of observations. For example:

```
double x[N] = {3.1, -1.2, 5.3, 1, 4.4, 21, 3, 7, -1.2, 3.2};
```

The corresponding program output should be:

```
Sorted data:
-1.200000 -1.200000 1.000000 3.000000 3.100000 3.200000 4.400000 5.300000 7.000000 21.000000

Median:
 3.150000
```