Solutions to Assignment #2 STA410H1F/2102H1F

1. 5. (a) The rejection algorithm accepts a proposal X from g if

$$U \le \frac{f(X)}{Mg(X)}$$

where $M = \max_x f(x)/g(x)$ and $X = b + Y/b$ with Y an exponential random variable with mean 1. Note that

$$\frac{f(x)}{g(x)} = k(b)\exp\left\{-\frac{1}{2}\left(x^2 - 2bx\right)\right\}$$

(where $k(b)$ is constant that is independent of x) is maximized at $x = b$ with

$$M = \frac{\exp(-b^2/2)}{\sqrt{2\pi}b(1-\Phi(b))}.$$

(To see that $f(x)/g(x)$ is maximized at $x = b$, note that the derivative of $\ln f(x) - \ln g(x)$ is $b - x$, which equals 0 at $x = b$ and is negative for $x > b$; thus $f(x)/g(x)$ is a decreasing function of x for $x \ge b$ and is maximized at $x = b$.) Thus

$$\frac{f(x)}{Mg(x)} = \exp\left\{-\frac{1}{2}(x-b)^2\right\}.$$

Therefore, we accept $X = b + Y/b$ if

$$\begin{aligned} U &\le \exp\left(-\frac{Y^2}{2b^2}\right) \\ \text{or} \quad -2\ln(U) &\ge \frac{Y^2}{b^2}. \end{aligned}$$

(b) For rejection sampling, the probability of acceptance is $1/M$ where M was evaluated in part (a). If we evaluate M for increasing values of b, it seems that this probability tends to 1. This can be verified using L'Hôpital's rule noting that $1/M = (1-\Phi(b))/(\phi(b)/b)$ where ϕ is the $\mathcal{N}(0,1)$ density:

$$\begin{aligned} \lim_{b\to\infty} \frac{1-\Phi(b)}{\phi(b)/b} &= \lim_{b\to\infty} \frac{\frac{d}{db}(1-\Phi(b))}{\frac{d}{db}[\phi(b)/b]} \\ &= \lim_{b\to\infty} \frac{-\phi(b)}{-\phi(b) - \phi(b)/b^2} \\ &= \lim_{b\to\infty} \frac{1}{1+b^{-2}} \\ &= 1. \end{aligned}$$

(An aside: for large values of b, we can approximate the tail probability $1 - \Phi(b)$ by Mills's ratio:

$$1 - \Phi(b) \approx \frac{1}{b}\phi(b).$$

This very useful approximation (and some refinements) can be proved by integrating

$$1 - \Phi(b) = \int_b^\infty \phi(x)\,dx = \int_b^\infty \frac{1}{x}\{x\phi(x)\}\,dx$$

by parts.)

(c) The "maximin" formulation is (probably) the best approach here. First of all, for a fixed value of $x \geq b$, we need to minimize $f(x)/g_\lambda(x)$ (or equivalently $\ln f(x) - \ln g_\lambda(x)$) over $\lambda > 0$. Using calculus, the minimizing value is $\lambda(x) = (x-b)^{-1}$. Then substituting $\lambda(x)$ for λ into $\ln f(x) - \ln g_\lambda(x)$, we find that (again using calculus) $\ln f(x) - \ln g_{\lambda(x)}(x)$ is maximized at

$$x_0(b) = \frac{b}{2} + \frac{1}{2}\sqrt{b^2+4}$$

and

$$\lambda(b) = \lambda(x_0(b)) = \frac{1}{x_0(b) - b} = x_0(b).$$

When $b = 0$ (Half-normal distribution), we have $\lambda(b) = 1$ while for large values of b, we have $\lambda(b) \approx b + 2$. Note that this method works even if $b \leq 0$.

2. (a) There are two ways to do this. The first (and simplest) is to note that when $y_i = a \times i + b$ then for $\theta_i = y_i$, we have

$$\sum_{i=1}^n (y_i - \theta_i)^2 + \lambda \sum_{i=2}^{n-1} (\theta_{i+1} - 2\theta_i + \theta_{i-1})^2 = 0$$

since $\theta_{i+1} - 2\theta_i + \theta_{i-1} = 0$. Since the objective is non-negative, $\widehat{\boldsymbol{\theta}} = (y_1, \cdots, y_n)$ must minimize it.

Alternatively, we can differentiate the objective function with respect to $\theta_1, \cdots, \theta_n$:

$$\frac{\partial}{\partial\theta_j}\left\{\sum_{i=1}^n (y_i - \theta_i)^2 + \lambda \sum_{i=2}^{n-1} (\theta_{i+1} - 2\theta_i + \theta_{i-1})^2\right\}$$
$$= \begin{cases} -2(y_1 - \theta_1) + 2\lambda(\theta_3 - 2\theta_2 + \theta_1) \ \text{ if } j = 1 \\ -2(y_2 - \theta_2) - 4\lambda(\theta_3 - 2\theta_2 + \theta_1) + 2\lambda(\theta_4 - 2\theta_3 + \theta_2) \ \text{ if } j = 2 \\ -2(y_j - \theta_j) + 2\lambda(\theta_{j+2} - 2\theta_{j+1} + +\theta_j) - 4\lambda(\theta_{j+1} - 2\theta_j + +\theta_{j-1}) + 2\lambda(\theta_j - 2\theta_{j-1} + +\theta_{j-2}) \\ \qquad \text{if } j = 3, \cdots, n-2 \\ -2(y_{n-1} - \theta_{n-1}) + 2\lambda(\theta_n - 2\theta_{n-1} + \theta_{n-2}) + 2\lambda(\theta_{n-1} - 2\theta_{n-2} + \theta_{n-3}) \ \text{ if } j = n-1 \\ -2(y_n - \theta_n) + 2\lambda(\theta_n - 2\theta_{n-1} + \theta_{n-2}) \ \text{ if } j = n \end{cases}$$

Setting these partial derivatives to 0, it follows that $\boldsymbol{y} = A\widehat{\boldsymbol{\theta}}$ where

$$A_\lambda = \begin{pmatrix} 1+\lambda & -2\lambda & \lambda & 0 & 0 & 0 & \cdots & 0 \\ -2\lambda & 1+5\lambda & -4\lambda & \lambda & 0 & 0 & \cdots & 0 \\ \lambda & -4\lambda & 1+6\lambda & -4\lambda & \lambda & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots & \\ 0 & 0 & 0 & \cdots & \lambda & -4\lambda & 1+5\lambda & -2\lambda \\ 0 & 0 & 0 & \cdots & 0 & \lambda & -2\lambda & 1+\lambda \end{pmatrix}.$$

It is straightforward to verify that

$$A_\lambda \begin{pmatrix} a+b \\ 2a+b \\ \vdots \\ na+b \end{pmatrix} = \begin{pmatrix} a+b \\ 2a+b \\ \vdots \\ na+b \end{pmatrix}.$$

(Note that this implies that $(a+b, 2a+b, \cdots, na+b)^T$ is an eigenvector of A with eigenvalue 1; in order words, this smoothing method preserves linear functions.)

(b) $\boldsymbol{y}^*$ and X are given by

$$\begin{aligned} \boldsymbol{y}^* &= \begin{pmatrix} y_1 \\ \vdots \\ y_n \\ 0 \\ \vdots \\ 0 \end{pmatrix} \\ X &= \begin{pmatrix} I \\ \sqrt{\lambda} B \end{pmatrix} \end{aligned}$$

where the (i, j) element of B is

$$b_{ij} = \begin{cases} 1 & \text{if } j = i \\ -2 & \text{if } j = i+1 \\ 1 & \text{if } j = i+2 \end{cases}$$

for $i = 1, \cdots, n-2$. Alternatively, we could also have

$$b_{ij} = \begin{cases} -1 & \text{if } j = i \\ 2 & \text{if } j = i+1 \\ -1 & \text{if } j = i+2 \end{cases}$$

for $i = 1, \cdots, n-2$.

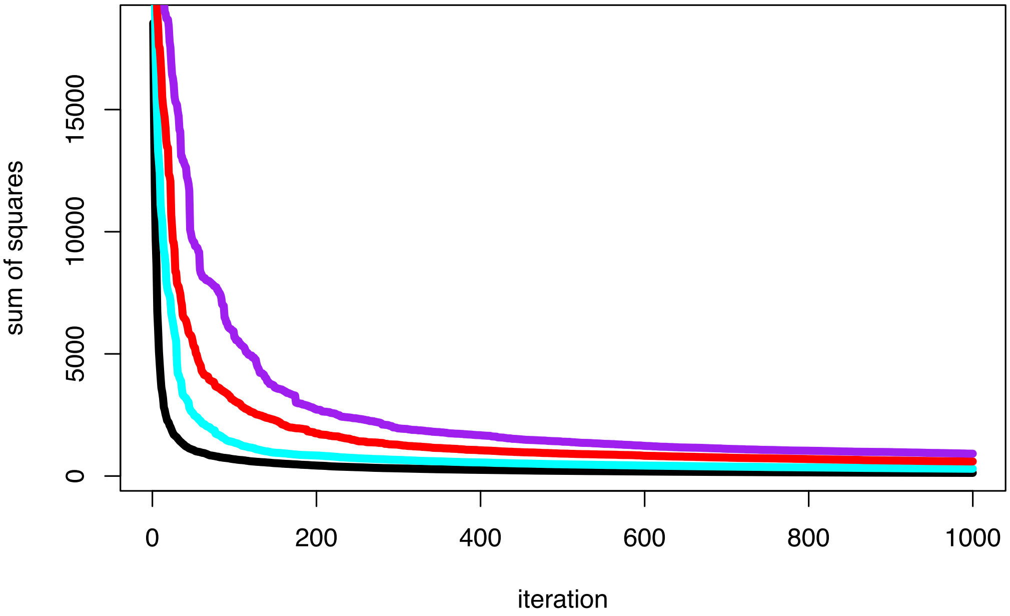

Figure 1: Objective function value versus iteration for $p = 5$ (purple), $p = 10$ (red), $p = 20$ (cyan), and $p = 50$ (black).

(c) Let w be the subset of parameters chosen at step $k+1$ of the algorithm and $\bar{w}$ to be its complement. Define $\widehat{\boldsymbol{\theta}}_w^{(k)}$ and $\widehat{\boldsymbol{\theta}}_{\bar{w}}^{(k)}$ to be the current estimates after k iterations of the algorithm with the value of the objective function after k iterations given by

$$\left\| \boldsymbol{y}^* - X_{\bar{w}} \widehat{\boldsymbol{\theta}}_{\bar{w}}^{(k)} - X_w \widehat{\boldsymbol{\theta}}_w^{(k)} \right\|^2 .$$

Since $\widehat{\boldsymbol{\theta}}_w^{(k+1)}$ minimizes

$$\left\| \boldsymbol{y}^* - X_{\bar{w}} \widehat{\boldsymbol{\theta}}_{\bar{w}}^{(k)} - X_w \boldsymbol{\theta}_w \right\|^2$$

with respect to $\boldsymbol{\theta}_w$, we have

$$\left\| \boldsymbol{y}^* - X_{\bar{w}} \widehat{\boldsymbol{\theta}}_{\bar{w}}^{(k)} - X_w \widehat{\boldsymbol{\theta}}_w^{(k+1)} \right\|^2 \leq \left\| \boldsymbol{y}^* - X_{\bar{w}} \widehat{\boldsymbol{\theta}}_{\bar{w}}^{(k)} - X_w \widehat{\boldsymbol{\theta}}_w^{(k)} \right\|^2$$

and so the objective function cannot increase from one iteration to the next.

(d) A plot of the objective function value versus iteration for $p = 5, 10, 20$, and 50 is given in Figure 2. Note that as p increases, the objective function value decreases more quickly as a function of the number of iterations. This is to be expected — at a given step, the more parameters over which we minimize the objective function, the smaller the minimized objective function will be.

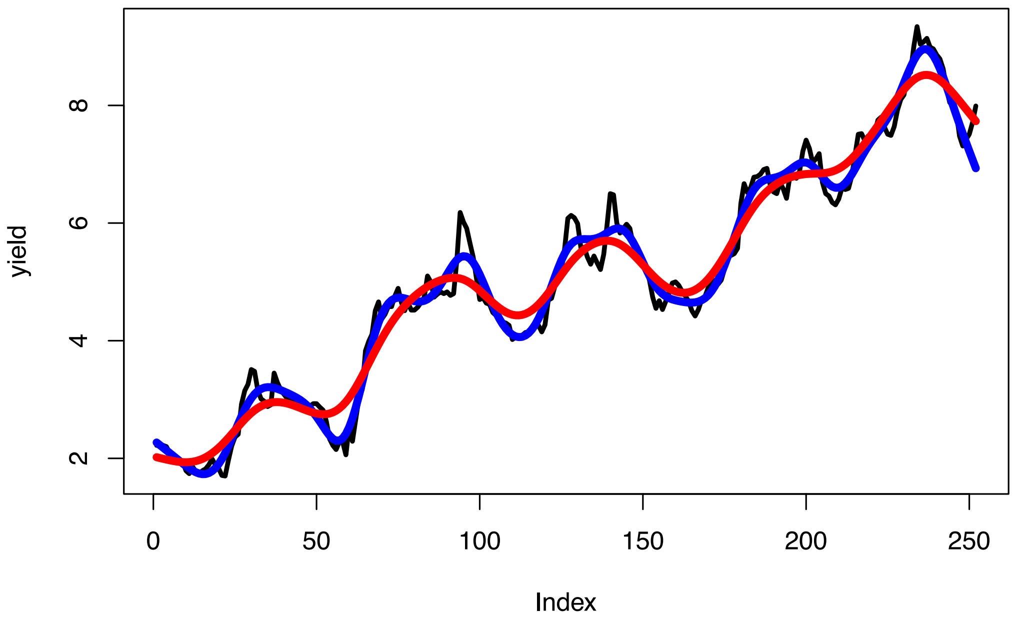

Figure 2: Original data (black) with randomized block G-S estimate for $p = 50$ after 1000 iterations (blue) and true minimizer (red)

(e) In part (d), we are trying to estimate $n = 252$ parameters by minimizing the objective function p parameters at a time. In choosing p, there is potentially a trade-off: The number of floating point operations needed at each iteration increases like p^2 while the objective function decreases more rapidly as we increase p. In this particular example, the optimal strategy is actually to set $p = n = 252$, in which case we only need one iteration to find the (exact) solution. However for larger n, this may not be the case. Figure 3 shows the original data with two estimates: the true minimizer (red curve) and the estimate using $p = 50$ after 1000 iterations; note that these two estimates differ considerably.

Supplemental problems

3. (a) The joint distribution of (V, Y) can be described in terms of the distribution of U as follows:

$$\begin{aligned}
P(V \le x, Y = k) &= P(k \le nU \le k + x) \\
&= P(k/n \le U \le (k+x)/n) \\
&= \frac{k+x}{n} - \frac{k}{n} = \frac{x}{n}.
\end{aligned}$$

for $0 < x < 1$ and $k = 0, \cdots, n-1$. Therefore, the marginal distribution of V is

$$P(V \le x) = \sum_{k=0}^{n-1} P(V \le x, Y = k) = x$$

(and so $V \sim \text{Unif}(0,1)$) while

$$P(Y = k) = P(V \le 1, Y = k) = \frac{1}{n}$$

(and so Y is uniformly distributed on $\{0, 1, \cdots, n-1\}$). For independence, note that

$$P(V \le x, Y = k) = \frac{x}{n} = P(V \le x)P(Y = k)$$

for $0 < x < 1$ and $k = 0, \cdots, n-1$.

(b) The method outlined in part (a) is very attractive as it allows us to generate two independent random variables for the price of one. However, as n increases, the number of digits in V decreases. Suppose that U has d decimal places so that

$$U = \sum_{j=1}^{d} M_j \times 10^{-j}$$

(where the digits $M_1, \cdots, M_d$ take values from 0 to 9). Suppose that $n = 10^k$; then

$$nU = \sum_{j=1}^{d} M_j \times 10^{k-j} = \sum_{\ell=0}^{k-1} M_{k-\ell} \times 10^{\ell} + \sum_{j=1}^{d-k} M_{j+k} \times 10^{-j}$$

and so

$$V = \lfloor nU \rfloor = \sum_{j=1}^{d-k} M_{j+k} \times 10^{-j}.$$

Thus if k is large compared d then the lost precision means that the distribution of V need not be close to a uniform distribution on $[0, 1]$.

4. (a) Define X, U, and X^* to be independent random variables where X has density g, $U \sim \text{Unif}(0,1)$, and X^* has density f_2^*. The method will return X if $U \le f_1(X)/g(X)$ and X^* otherwise. Therefore the distribution function G of the random variable is

$$G(x) = P(U \le f_1(X)/g(X), X \le x) + P(U > f_1(X)/g(X), X^* \le x).$$

For the first term on the right hand side above, we have

$$P(U \le f_1(X)/g(X), X \le x) = \int_{-\infty}^{x} \int_0^{f_1(t)/g(t)} g(t)\, du\, dt = \int_{-\infty}^{x} f_1(t)\, dt$$

while the second term is

$$\begin{aligned} P(U > f_1(X)/g(X), X^* \le x) &= P(U > f_1(X)/g(X)) P(X^* \le x) \\ &= \left\{ \int_{-\infty}^{\infty} \int_{f_1(x)/g(x)}^{1} g(x)\, du\, dx \right\} \left\{ \int_{-\infty}^{x} f_2^*(t)\, dt \right\} \\ &= \left\{ 1 - \int_{-\infty}^{\infty} f_1(t)\, dt \right\} \int_{-\infty}^{x} f_2^*(t)\, dt \\ &= \left(\int_{-\infty}^{\infty} f_2(t)\, dt \right) \int_{-\infty}^{x} f_2^*(t)\, dt \\ &= \int_{-\infty}^{x} f_2(t)\, dt \end{aligned}$$

Therefore

$$G(x) = \int_{-\infty}^{x} f_1(t)\, dt + \int_{-\infty}^{x} f_2(t)\, dt = \int_{-\infty}^{x} f(t)\, dt.$$

The probability that the X generated in step 1 is rejected in step 2 is

$$P(U > f_1(X)/g(X)) = \int_{-\infty}^{\infty} f_2(t)\, dt$$

(b) Since $f_2(x) = k$ and $g(x) = 1/2$, we must have $f_1(x) = f(x) - k \le 1/2$ for all x. f is maximized at $x = 0$ with $f(0) = 2/\pi$ and so $k \ge 2/\pi - 1/2 \approx 0.137$. Likewise, $k \le f(\pm 1) = 1/\pi \approx 0.318$. So the A-C method can be applied here (assuming g and f_2^* uniform) for $2/\pi - 1/2 \le k \le 1/\pi$.
(c) The probability of rejection of the proposal from the uniform density g is $\int_{-1}^{1} f_2(t)\, dt = 2k$. This is minimized at $k = 2/\pi - 1/2$ with the probability of rejection equal to $4/\pi - 1 \approx 0.273$. (As a means of comparison, if we use rejection sampling with a uniform proposal, the expected number of uniform random variables generated is $2 \times 4/\pi \approx 2.546$ while for the best A-C method, we require $2 + (4/\pi - 1) \approx 2.273$ uniforms.)

5. (a) Define $\phi(u, v) = (u, v/u) = (w, x)$. The inverse of ϕ is $\phi^{-1}(w, x) = (w, x\, w)$ and the Jacobian of the inverse is

$$J_{\phi^{-1}}(w, x) = \left| \det \begin{pmatrix} 1 & x \\ 0 & w \end{pmatrix} \right| = w$$

and so the joint density of $(U, X) = (W, X)$ is

$$g(u, x) = \frac{u}{|\mathcal{C}_h|} \quad \text{for } 0 \le u \le \sqrt{h(x)}$$

and the marginal density of X is

$$f_X(x) = \int_0^{\sqrt{h(x)}} \frac{u}{|\mathcal{C}_h|}\, du = \frac{h(x)}{2|\mathcal{C}_h|}.$$

From this, it follows that

$$|\mathcal{C}_h| = \frac{1}{2}\int_{-\infty}^{\infty} h(x)\, dx.$$

(b) $\mathcal{C}_h = \left\{(u,v) : 0 \le u \le \sqrt{h(v/u)}\right\}$. If $(u,v) \in \mathcal{C}_h$ then $u \le \max_x \sqrt{h(x)}$ and so u must lie in the interval $[0, \max_x \sqrt{h(x)}]$. Since u is positive, we must have

$$\frac{1}{u}\sqrt{h(v/u)} \ge 1$$

and so if $v > 0$, we have

$$v \le \frac{v}{u}\sqrt{h(v/u)} \le \max_x x\sqrt{h(x)} = v_+.$$

On the other hand, if $v < 0$,

$$v \ge \frac{v}{u}\sqrt{h(v/u)} \ge \min_x x\sqrt{h(x)} = v_-.$$

Therefore $(u,v) \in \mathcal{C}_h$ must lie in the rectangle $[0, u_+] \times [v_-, v_+]$.
(c) The function will look something like the following:

```
rnormal <- function(n) {
               x <-  NULL
               rejections <- 0 # we will count the number of rejections
               vbound <- sqrt(2/exp(1))
               for  (i in 1:n) {
                  reject <- T
                  while (reject) {
                     u <- runif(1) # single Unif(0,1) rv
                     v <- runif(1,-vbound,vbound)
                     # rejection sampling test
                     if (u<=exp(-(v/u)^2/4)) {
                        x <- c(x, v/u)
                        reject <- F
                        }
                     else rejections <- rejections +1
                     }
                  }
                accept.rate <- n/(n+rejections)
```

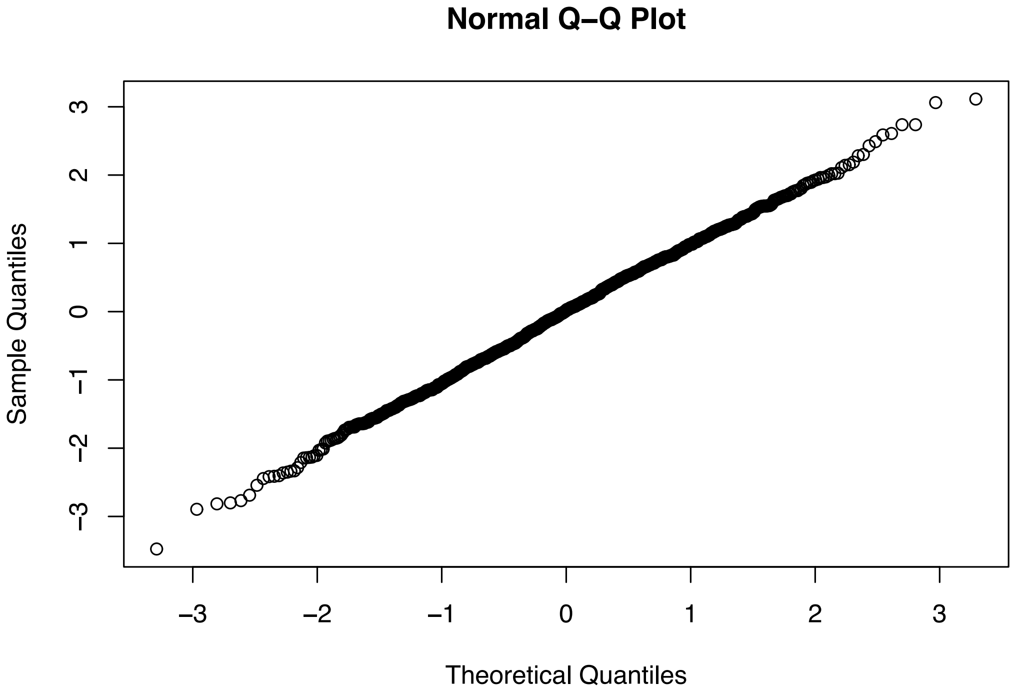

Figure 3: Normal quantile-quantile plot of output from `rnormal`.

```
        r <- list(x=x,accept.rate=accept.rate)
        r
        }
```

Note that this function estimates the acceptance rate for the rejection sampling; for example,

```
> r <- rnormal(1000)
> qqnorm(r$x)
> r$accept.rate
[1] 0.7102273
```

The normal quantile-quantile plot is shown in Figure 1. Note that in this case, it is straightforward to evaluate the acceptance rate as the ratio of $|\mathcal{C}_h|$ to the area of the approximating rectangle: $|\mathcal{D}_h| = 2 \times \sqrt{2/e}$. The area of $\mathcal{C}_h$ is simply

$$|\mathcal{C}_h| = \frac{1}{2}\int_{-\infty}^{\infty} \exp(-x^2/2)\,dx = \sqrt{\pi/2}$$

and so the theoretical rejection rate is $\sqrt{\pi/2}/(2\sqrt{2/e}) = 0.7306$.

6. (a) Using integration by parts repeatedly, we have

$$P(T_k \geq 1) \;\; = \;\; \int_1^{\infty} \frac{\lambda^k x^{k-1}\exp(-\lambda x)}{(k-1)!}\,dx$$

$$
\begin{aligned}
&= \frac{\lambda^{k-1}\exp(-\lambda)}{(k-1)!} + \int_1^\infty \frac{\lambda^{k-1}x^{k-2}\exp(-\lambda x)}{(k-2)!}\,dx \\
&\vdots \sum_{j=0}^{k-1} \frac{\lambda^j \exp(-\lambda)}{j!}.
\end{aligned}
$$

(b) Note that $P(T_k \geq 1) = P(N \leq k-1)$ where $N \sim \text{Poisson}(\lambda)$. Therefore, we could generate a Poisson random variable with mean λ by defining $T_0 = 0$ and $T_k = E_1 + \cdots + E_k$ for $k \geq 1$ and then defining

$$N = \{k : T_k < 1 \leq T_{k+1}\} = \max\{k : T_k < 1\}$$

since $[N \leq k-1] = [T_k \geq 1]$.