Stat 102C HW1: Answer Key

Jonathan Arfa, Muzhou Liang

May 7, 2014

Problem 1.1

$$
\begin{aligned}
P(Y \leq x) &= P(F^{-1}(U) \leq x) && (1)\\
&= P(U \leq F(x)) && (2)\\
&= \int_0^{F(x)} \mathrm{d}u && (3)\\
&= F(x) && (4)
\end{aligned}
$$

Problem 1.2

Textbook, p.44
$P(F(X) \leq u) = P[X \leq F^{-1}(u)] = F(F^{-1}(u)) = u$

Problem 2.1

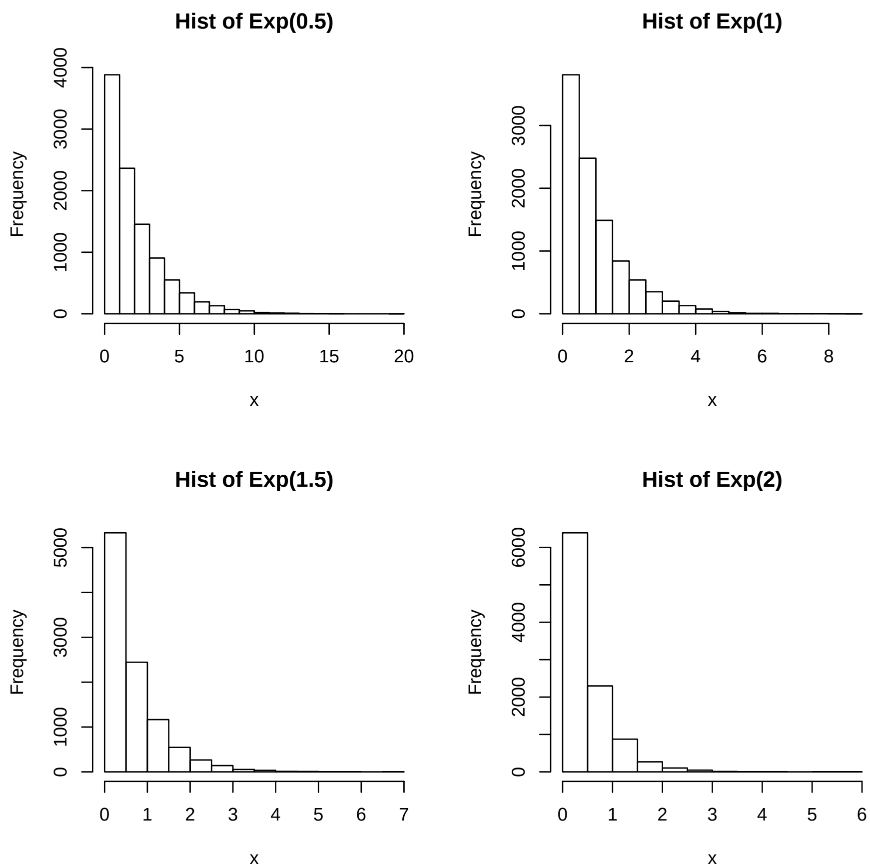
The pdf is: $f(x) = \lambda exp(\text{-}\lambda x)$
So, the CDF is: $F(x) = \int_0^x f(x)\,\mathrm{d}x = 1 - \exp(\text{-}\lambda x)$

Problem 2.2

$u = 1 - \exp(\text{-}\lambda x)$
$1 - u = \exp(\text{-}\lambda x)$
$\log(1 - u) = -\lambda x$
$x = -\log(1 - u)/\lambda$
Since $u \in [0, 1]$, this is eqiuvalent to $x = -\log(u)/\lambda$. Isn't that prettier?

Problem 2.3 and 2.4

```r
exp_invcdf = function(n, lambda) {
    u = runif(n)
    return(-log(u)/lambda)
}
par(mfrow = c(2, 2))
lambda_nums = seq(0.5, 2, 0.5)
n = 10000
for (lambda in lambda_nums) {
    hist(exp_invcdf(n, lambda), xlab = "x", main = paste0("Hist of Exp(",
        lambda, ")"))
}
```



Problem 3.1

This is identical to exercise 2.5. Because X is generated from g, so we integral over g.

$$\begin{aligned} P(U < f(X)/Mg(X)) &= \int_{-\infty}^{\infty} \frac{f(x)}{Mg(x)} g(x)\mathrm{d}x && (5) \\ &= \int_{-\infty}^{\infty} \frac{f(x)}{M} \mathrm{d}x && (6) \\ &= \frac{1}{M} && (7) \end{aligned}$$

Problem 3.2

See p.52 in your textbook.

Problem 4.1

I'll tell you the real way to prove this, and the intuitive explanation.
Proof:

$$\begin{aligned} P(X < x | X > c) &= \frac{P(X < x, X > c)}{P(X > C)} && (8) \\ F(x) &= \frac{\int_C^x \phi(x)\,\mathrm{d}x}{1 - \Phi(C)} && (9) \\ f(x) &= \frac{d}{dx}\left[\frac{\int_C^x \phi(x)\,\mathrm{d}x}{1 - \Phi(C)}\right] && (10) \\ &= \frac{\phi(x)}{1 - \Phi(C)} && (11) \end{aligned}$$

Intuitive: The shape of PDF over the interval $[C, \infty]$ remains the same as it was with $N(0,1)$, but now the probability outside of that range is 0. So the pdf over that range must be shifted upward by some constant.

What is that constant? Remember that the area under the new pdf needs to sum to one. $\Phi(x)$ denotes the area under $N(0,1)$ from 0 to x, so $1 - \Phi(x)$ denotes the area under $N(0,1)$ from x to ∞. Since $\phi(x)$ is the pdf of $N(0,1)$, the new pdf is: $f(x) = \phi(x)/(1 - \Phi(C))$ for $x \geq C > 0$

Problem 4.2

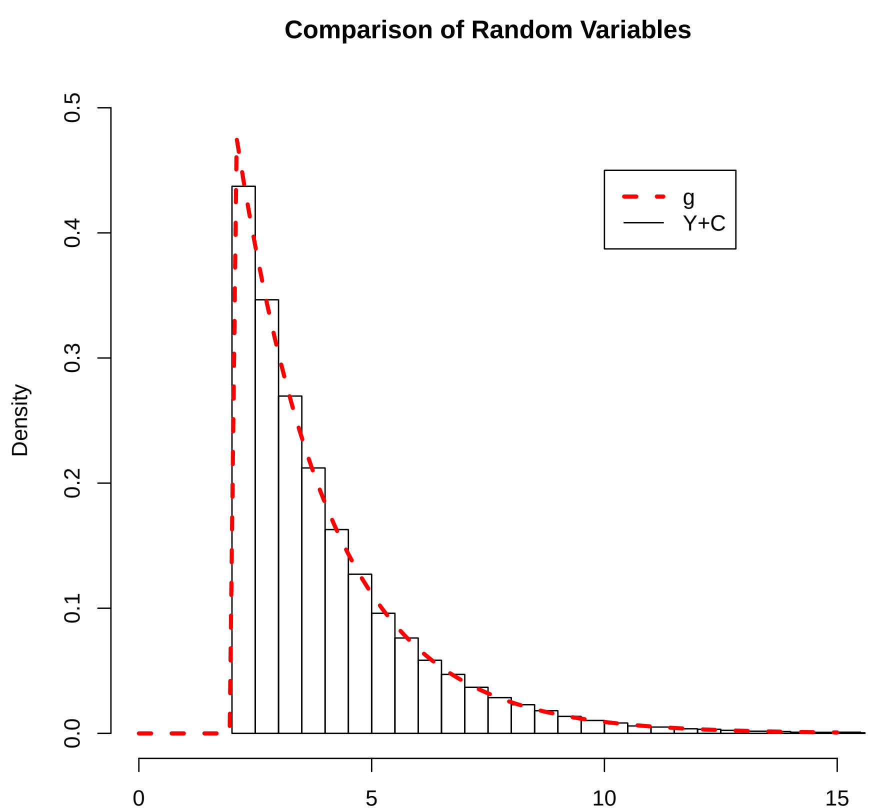
$$G_X(x) = \int_C^x g(x)\mathrm{d}x = 1 - \exp[-\lambda(x - C)] \quad (12)$$

$$G_Y(y) = P(Y < y) = P(X < y + C) = G_X(y + C) = 1 - \exp(-\lambda y) \quad (13)$$

Hence, $Y \sim \mathcal{E}xp(\lambda)$.

Just to show this graphically...

```
g = function(x, lambda = 0.5, C = 2) {
    return(ifelse(x >= C, lambda * exp(-lambda * (x -
        C)), 0))
}
hist(rexp(n = 1e+05, rate = 0.5) + 2, freq = FALSE,
    breaks = 50, xlim = c(0, 15), ylim = c(0, 0.5),
    main = "Comparison of Random Variables", xlab = "")
curve(g, from = 0, to = 15, lty = 2, lwd = 3, col = "red",
    add = TRUE)
legend(10, 0.45, legend = c("g", "Y+C"), lwd = c(3,
    1), lty = c(2, 1), col = c("red", "black"))
```



Problem 4.3

Let $\frac{\partial f(x)/g(x)}{\partial x} = 0$, we will get $x = \lambda$. Thus,

$$M(\lambda) = \frac{1}{\lambda\sqrt{2\pi}[1 - \Phi(C)]} \exp\{\frac{\lambda^2}{2} - \lambda C\}$$

Let $\frac{\partial M(\lambda)}{\partial \lambda} = 0$, we will get $\lambda = \frac{C+\sqrt{C^2+4}}{2}$.

As the code below shows, if $C = 1$ the optimal lambda $\lambda = 1.618$, and the M at that point is $M = 1.141$.

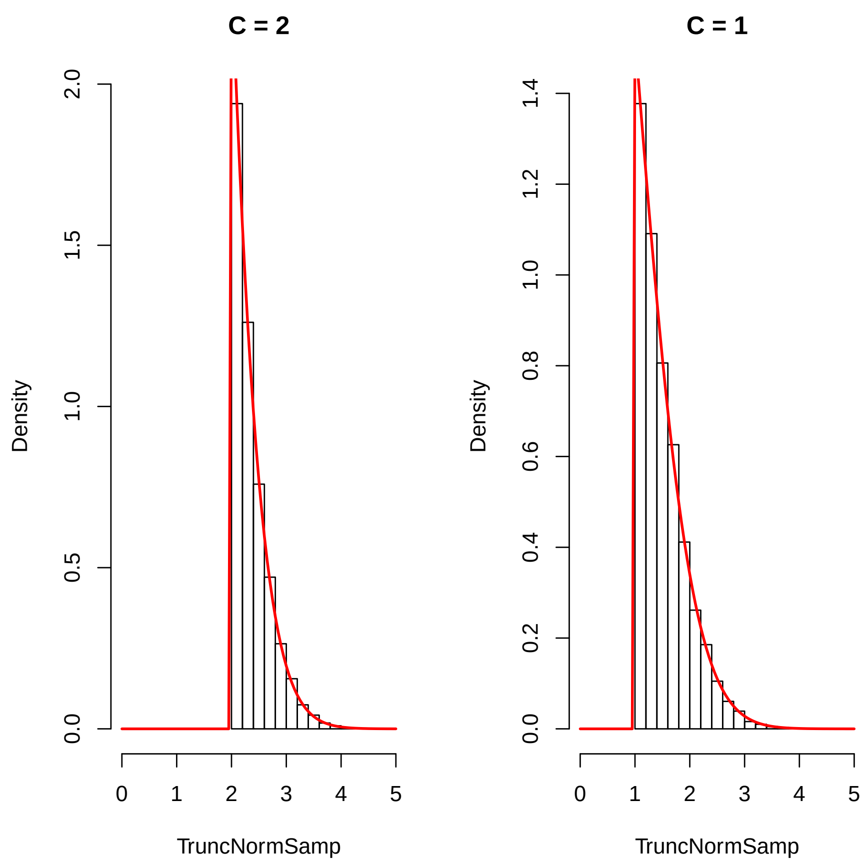
Problem 4.4

```
f = function(x, C) dnorm(x)/(1 - pnorm(C))

g = function(x, lambda, C) lambda * exp(-lambda * (x -
    C))

M = function(lambda, C) {
    f(lambda, C)/g(lambda, lambda, C)
}
```

```
samplefrom = function(C) {
    lambda = (C + sqrt(C^2 + 4))/2
    m = M(lambda, C)
    y = rexp(1, lambda)
    x = y + C
    u = runif(1)
    i = 1
    # u*M*g(x) > f(x)
    while (u * m * g(x, lambda, C) > f(x, C)) {
        y = rexp(1, lambda)
        x = y + C
        u = runif(1)
        i = i + 1
    }
    res = list(counter = i, sample = x)
    return(res)
}
```

Problem 4.5

```
Nsim = 10000
library(truncnorm)
par(mfrow = c(1, 2))

TruncNormSamp = replicate(Nsim, samplefrom(2)$sample)
hist(TruncNormSamp, freq = FALSE, main = "C = 2", xlim = c(0,
    5))
curve(dtruncnorm(x, a = 2), col = "red", lwd = 2, add = T)

TruncNormSamp = replicate(Nsim, samplefrom(1)$sample)
hist(TruncNormSamp, freq = FALSE, main = "C = 1", xlim = c(0,
    5))
curve(dtruncnorm(x, a = 1), col = "red", lwd = 2, add = T)
```

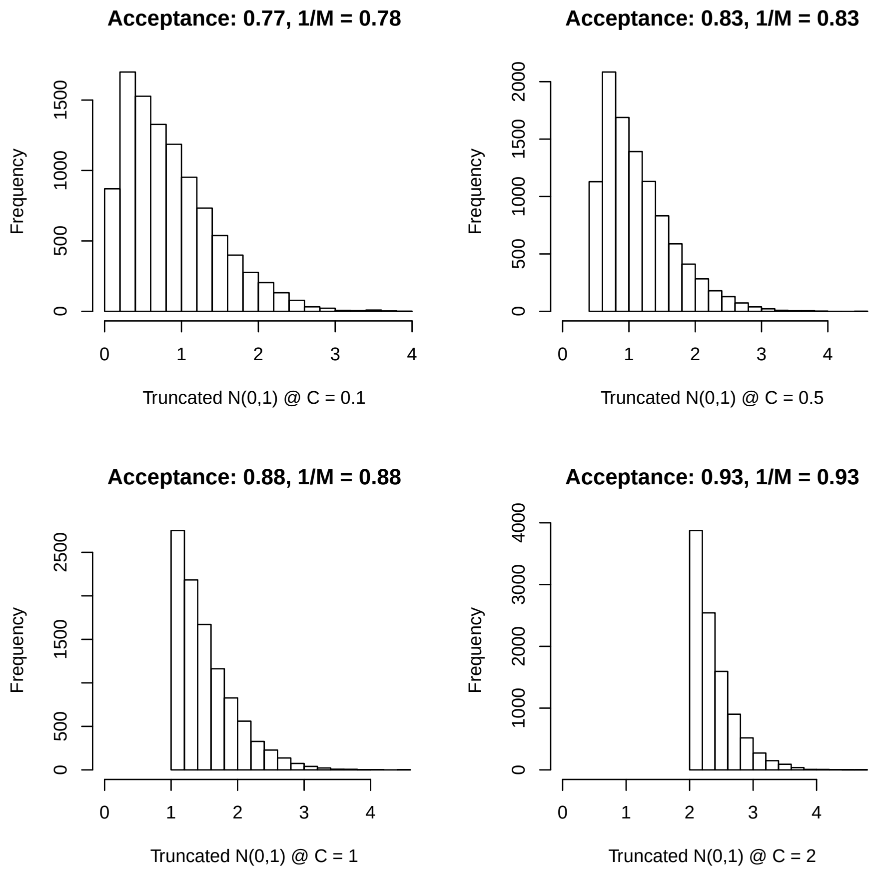


```r
rate = function(Nsim, C) {
    j = 0
    freq = 0
    for (j in 1:Nsim) freq = freq + samplefrom(C)$counter
    lambda = (C + sqrt(C^2 + 4))/2
    m = M(lambda, C)
    res = list(rate = Nsim/freq, prob = 1/m)
    return(res)
}

rate(Nsim, 2)  # see acceptance rate when C = 2

## $rate
## [1] 0.9347
##
## $prob
## [1] 0.9336

rate(Nsim, 1)  # see acceptance rate when C = 2

## $rate
## [1] 0.8816
##
## $prob
## [1] 0.8765
```

An alternative way for Problem 4.4 and 4.5

```r
f = function(x, C) {
    ifelse(x >= C, dnorm(x)/(1 - pnorm(C)), 0)
}
g = function(x, lambda, C) {
    # ifelse(x >=C, lambda * exp(-lambda * (x-C)), 0)
    ifelse(x >= C, dexp((x - C), lambda), 0)
}
M_funct = function(lambda, C) {
    optimize(f = function(x) {
        f(x, C)/g(x, lambda, C)
    }, interval = c(C, C + (20 * lambda)), maximum = TRUE)$objective
}
gen_truncNorm = function(nsamples, Mfunct, C = 1) {
    # finds lambda and M
    optimum = optimize(f = function(L) M_funct(L, C),
        interval = c(0.1, 7), maximum = FALSE)
    lambda = optimum$minimum
    M_min = optimum$objective
```

```r
    nsim = nsamples
    # start with num_samples, add 1 for each additional
    # simulation
    target = numeric(nsamples)
    for (i in 1:nsamples) {
        u = runif(1) * M_min
        y = rexp(1, lambda) + C
        while (u > f(y, C)/g(y, lambda, C)) {
            u = runif(1) * M_min
            y = rexp(1, lambda) + C
            nsim = nsim + 1
        }
        target[i] = y
    }
    return(list(x = target, accept = nsamples/nsim,
        M = M_min))
}

N = 10000
Cvals = c(0.1, 0.5, 1, 2)
par(mfrow = c(2, 2))
for (Cstar in Cvals) {
    truncNorm = gen_truncNorm(N, M_funct, C = Cstar)
    hist(truncNorm$x, xlim = c(0, max(truncNorm$x)),
        xlab = paste0("Truncated N(0,1) @ C = ", Cstar),
        main = paste0("Acceptance: ", round(truncNorm$accept,
            2), ", 1/M = ", round(1/truncNorm$M, 2)))
}
```


Acceptance: 0.77, 1/M = 0.78
Frequency
0
500
1000
1500
0
1
2
3
4
Truncated N(0,1) @ C = 0.1
Acceptance: 0.83, 1/M = 0.83
Frequency
0
500
1000
1500
2000
0
1
2
3
4
Truncated N(0,1) @ C = 0.5
Acceptance: 0.88, 1/M = 0.88
Frequency
0
500
1500
2500
0
1
2
3
4
Truncated N(0,1) @ C = 1
Acceptance: 0.93, 1/M = 0.93
Frequency
0
1000
2000
3000
4000
0
1
2
3
4
Truncated N(0,1) @ C = 2