QBS 120 - Problem Set 3

Rob Frost

Grading: Problems 3 and 8, 5 pts each; see below for grading details

1. **(Based on Rice, Chapter 4, Problem 54) Let X, Y, and Z be independent RVs with variances $\sigma_X^2, \sigma_Y^2, \sigma_Z^2$. Let:**

$$U = Z - X$$
$$V = Z - Y$$

   (a) **Find Cov(U, V) and ρ_{UV}.**
   We are asked to find the covariance and correlation coefficient for linear functions of independent RVs. Starting with the definition of the covariance of U and V, we'll plug in the linear expressions and simplify:

$$\begin{aligned}
Cov(U,V) &= E[UV] - E[U]E[V] && \text{by defn}\\
&= E[(Z-X)(Z-Y)] - E[Z-X]E[Z-Y] && \text{plug in U and V}\\
&= E[Z^2 - XZ - YZ + XY] - E[Z-X]E[Z-Y] && \text{simplify}\\
&= E[Z^2] - E[XZ] - E[YZ] + E[XY]\\
&- (E[Z] - E[X])(E[Z] - E[Y]) && \text{E of sums}\\
&= E[Z^2] - E[Z]^2 - E[XZ] - E[YZ]\\
&+ E[XY] + E[X]E[Z] + E[Y]E[Z] - E[X]E[Y] && \text{simplfy}\\
&= Var(Z) - Cov(X,Z) - Cov(Y,Z) + Cov(X,Y) && \text{def var and cov}\\
&= \sigma_Z^2 && \text{Cov is 0 for independent RVs}
\end{aligned}$$

   To find ρ_{UV} we need Var(U) and Var(V). For Var(U):

$$\begin{aligned}
Var(U) &= Var(Z - X)\\
&= Var(Z) + Var(-X) && \text{Z and X uncor.}\\
&= Var(Z) + Var(X) && Var(bX) = b^2 Var(X)\\
&= \sigma_Z^2 + \sigma_X^2
\end{aligned}$$

   Following similar logic, $V(V) = \sigma_Z^2 + \sigma_Y^2$. By the definition of correlation:

$$\begin{aligned}
\rho_{UV} &= \frac{Cov(U,V)}{\sqrt{Var(U)Var(V)}}\\
&= \frac{\sigma_Z^2}{\sqrt{(\sigma_Z^2 + \sigma_X^2)(\sigma_Z^2 + \sigma_Y^2)}}
\end{aligned}$$

(b) **If $U = Z + X$ and $V = Z + Y$, do the values of $Cov(U, V)$ and $\rho_{U,V}$ computed in part a) change? Explain.**

The values remain the same. $Var(Z + X) = Var(Z - X)$ since $Var(bX) = b^2 Var(X)$

(c) **How does $\rho_{U,V}$ change if σ_Z^2 is much larger than σ_X^2 or σ_Y^2?**

If σ_Z^2 is much larger than σ_X^2 or σ_Y^2, the correlation will be dominated by Z.

(d) **How does $\rho_{U,V}$ change if σ_Z^2 is much smaller than σ_X^2 or σ_Y^2?**

If the reverse holds, the correlation will be close to 0, which makes sense given that U and V will be close to independent.

(e) **How do the answers for parts c) and d) relate to variable standardization?**

The standardization of variables prior to statistical analysis (e.g., predictors in a regression model) is motivated by similar logic, i.e., to prevent variables with large variances from dominating the solution.

2. **(Based on Rice, Chapter 4, Problem 64) Let X and Y be jointly distributed RVs with correlation ρ_{XY}; define the standardized random variables $\bar{X}$ and $\bar{Y}$ as:**

$$\bar{X} = (X - E[X])/\sqrt{Var(X)}$$
$$\bar{Y} = (Y - E[Y])/\sqrt{Var(Y)}$$

(a) **Show that $Cov(\bar{X}, \bar{Y}) = \rho_{XY}$.**

According to the definition of covariance:

$$\begin{aligned}
Cov(\bar{X}, \bar{Y}) &= E[\bar{X}\bar{Y}] - E[\bar{X}]E[\bar{Y}] \\
&= E\Big[\frac{(X - E[X])(Y - E[Y])}{\sqrt{Var(X)Var(Y)}}\Big] - E\Big[\frac{X - E[X]}{\sqrt{Var(X}}\Big]E\Big[\frac{Y - E[Y]}{\sqrt{Var(Y)}}\Big] \\
&= \frac{E[(X - E[X])(Y - E[Y])] - E[X - E[X]]E[Y - E[Y]]}{\sqrt{Var(X)Var(Y)}} && \text{E of const = const} \\
&= \frac{Cov(X, Y) - E[X - E[X]]E[Y - E[Y]]}{\sqrt{Var(X)Var(Y)}} && \text{def of cov} \\
&= \frac{Cov(X, Y) - (E[X] - E[X])(E[Y] - E[Y])}{\sqrt{Var(X)Var(Y)}} && \text{E of const = const} \\
&= \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}} && \text{simplify} \\
&= \rho_{XY} && \text{def of } \rho
\end{aligned}$$

(b) **Principal component analysis (PCA) is normally defined by the eigenvalue decomposition of the sample covariance matrix for multivariate data. Look at the R PCA function prcomp(). What is the impact of setting center=T and scale=T when calling prcomp()? When might this be desirable?**

When calling prcomp() with scale=T and center=T, we are standardizing the input data (though standardizing with estimates of the mean and SD rather than the population values as in this problem). The result is that the eigenvalue decomposition is performed on the sample correlation matrix rather than the sample covariance matrix (i.e., the sample covariance matrix computed on standardized data is, per this problem, the sample correlation matrix). This can be beneficial when we want to capture the pattern of correlation between variables rather than the pattern of covariance which, by definition, will be dominated by the variables with high variance.

3. **(Based on Rice, Chapter 4, Problem 74) The number of offspring of an organism is a discrete random variable with mean μ and variance σ^2. Each of its offspring reproduces in the same manner.**

(a) **Find the expected number of offspring in the third generation.**

Let the number of offspring in the second generation be represented by the random variable N. From the problem statement we know:

$$E[N] = \mu$$
$$Var(N) = \sigma^2$$

Let the number of offspring of each second generation organism be represented by the random variables $N2_1, ..., N2_N$. From the problem statement we know:

$$E[N2_i] = \mu$$
$$Var(N2_i) = \sigma^2$$

The total number of offspring in the third generation is given by a third random variable $T3$ defined as follows:

$$T3 = \sum_{i=1}^{N} N2_i$$

Goal is to find the expected number of offspring in third generation or $E[T3]$, which takes the following value per the Law of Total Expectation:

$$E[T3] = E_N[E_{T3}[T3|N]]$$

Grading: 1 pt to frame desired expectation using the law of total expectation .

If the initial number of offspring were fixed, $N = n$, $E[T3|N = n] = nE[N2_i]$. Therefore, for a variable number of initial offspring, $E[T3|N] = NE[N2_i]$.

$$\begin{aligned} E[T3] &= E[NE[N2_i]] \\ &= E[N]E[N2_i] \\ &= \mu\mu \\ &= \mu^2 \end{aligned}$$

Grading: 1 pt to find the correct expectation.

(b) **Find the variance of the number of offspring in the third generation.**

To find the variance of T, can use the formula:

$$Var(Y) = Var(E[Y|X]) + E[Var(Y|X)]$$

For this problem, the formula becomes:

$$Var(T3) = Var(E[T3|N]) + E[Var(T3|N)]$$

As specified above, $E[T3|N] = NE[N2_i]$. Because $Var(T3|N = n) = Var(\sum_{i=1}^{n} N2_i) = nVar(N2_i)$, $Var(T3|N) = NVar(N2_i)$. Given these, the variance of $T3$ can be found as follows:

$$
\begin{aligned}
Var(T3) &= Var(E[T3|N]) + E[Var(T3|N)] \\
&= Var(NE[N2_i]) + E[NVar[N2_i]] \text{ , } \textit{plug in values from above} \\
&= E[N2_i]^2 Var(N) + Var[N2_i]E[N] \text{ , } \textit{move out constants} \\
&= \mu^2\sigma^2 + \sigma^2\mu \text{ , } \textit{plug in values} \\
&= \mu\sigma^2(\mu + 1)
\end{aligned}
$$

Grading: 0.5 pts to use the correct variance equation; 0.5 pts to find the correct variance.

(c) **Validate your answers to a) and b) via simulation with the number of offspring represented by a Poisson RV with $\lambda = 2$. Create 1000 separate populations that each include 3 generations and use a histogram to visualize the empirical distribution of the number of offspring in the third generation. Estimate the expected number of 3rd generation offspring using the average across all 1000 simulations and estimate the variance of the number using the R var() function (we will learn the basis for these estimates in Chapter 8). Compare these estimates with the values computed according to the results in part a) and b).**

First, let's write a function that simulates the number of members in the third generation for a single population:

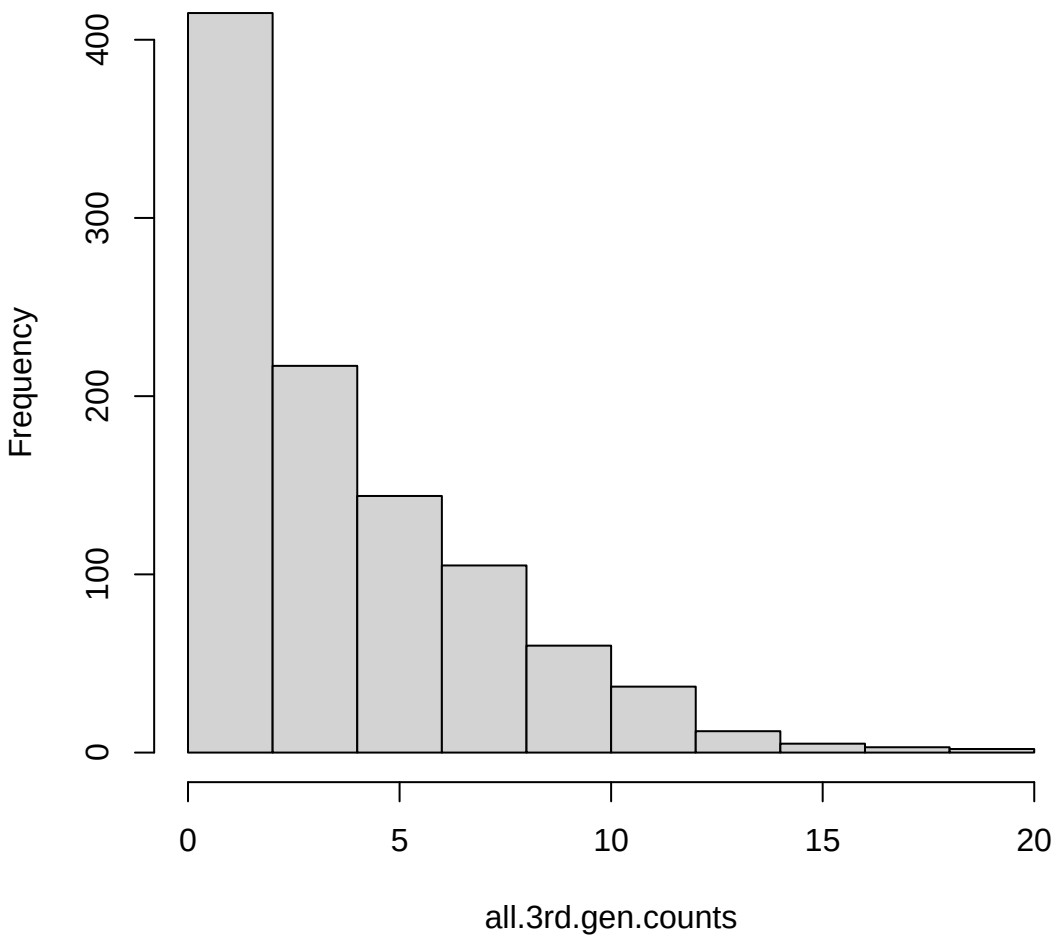
```
> simThirdGen = function() {
+ # simulate 2 generation
+ num.2 = rpois(n=1,lambda=2)
+ # For each member of the second, simulate children
+ num.3=0
+ if (num.2 == 0) {
+         return (0)
+ }
+ for (i in 1:num.2) {
+         num.3 = num.3 + rpois(n=1,lambda=2)
+ }
+ return (num.3)
+ }
> simThirdGen()
[1] 2
```

Simulate for 1000 separate populations:

```
> num.sims=1000
> all.3rd.gen.counts = rep(0, num.sims)
> for (i in 1:num.sims) {
+         all.3rd.gen.counts[i] = simThirdGen()
+ }
```

Plot empirical distribution as a histogram:

```
> plot(hist(all.3rd.gen.counts))
```



Estimate the expectation and variance of number of members in the 3rd generation:

```
> (est.exp = mean(all.3rd.gen.counts))
[1] 3.985
> (est.var = var(all.3rd.gen.counts))
[1] 12.58936
```

To compare with the results from parts a) and b) we note that the variance and expected value for a Poisson RV are both $\lambda = 2$.

$$\begin{aligned} E[T3] &= \mu^2 \\ &= \lambda^2 \\ &= 4 \end{aligned}$$

$$\begin{aligned} Var[T3] &= \mu\sigma^2(\mu + 1) \\ &= \lambda\lambda(\lambda + 1) \\ &= 2 * 2(2 + 1) \\ &= 12 \end{aligned}$$

The estimates are fairly close to the true values. Since the estimates are themselves RVs, we expect some variance around their expected values (which equal the true values of 4 and 12). This type of validation via simulation can be a useful tool for checking complex analytical calculations.

Grading: 1 pt to get simulation-based estimates; 1 pt if the estimates look valid, i.e., are close to the true values.

4. **(Optional - Rice, Chapter 4, Problem 81) Find the moment-generating function of a Bernoulli RV and use it to find the mean, variance and third central moment.**

Let X be the Bernoulli random variable with parameter p. The mgf can be computed from the expectation-based definition as follows:

$$\begin{aligned}
M_X(t) &= E[e^t X] \\
M_X(t) &= \sum_{x \in 0,1} e^t x p^x (1-p)^{1-x} \\
M_X(t) &= e^t 0 p^0 (1-p)^{1-0} + e^t 1 p^1 (1-p)^{1-1} \\
M_X(t) &= 1 - p + p e^t
\end{aligned}$$

To compute the first, second and third moments of X from the mgf, it is necessary to compute the first, second and third derivative of the mgf with respect to t:

$$\begin{aligned}
M_X^{'}(t) &= p e^t \\
M_X^{''}(t) &= p e^t \\
M_X^{'''}(t) &= p e^t
\end{aligned}$$

The mean of X can be computed from the mgf as follows:

$$E[X] = M_X^{'}(0) = p$$

The variance of X can be computed from the mgf as follows:

$$\begin{aligned}
Var(X) &= E[X^2] - E[X]^2 \\
Var(X) &= M_X^{''}(0) - M_X^{'}(0)^2 \\
Var(X) &= p - p^2 \\
Var(X) &= p(1-p)
\end{aligned}$$

The third central moment can be computed as follows:

$$\begin{aligned}
&E[(X-\mu)^3] \\
&= E[(X-p)^3] \\
&= E[(X^2 - 2pX + p^2)(X-p)] \\
&= E[X^3 - 3pX^2 + 3p^2X - p^3] \\
&= E[X^3] - E[3pX^2] + E[3p^2X] - E[p^3] \\
&= p - 3p^2 + 3p^3 - p^3 \\
&= p - 3p^2 + 2p^3 \\
&= p(1-p)(1-2p)
\end{aligned}$$

5. **(Based on Rice, Chapter 5, Problem 1) Let $X_1, X_2, ...$ be a sequence of independent random variables with $E[X_i] = \mu$ and $Var(X_i) = \sigma_i^2$. Show that if $n^{-2}\sum_{i=1}^{n}\sigma_i^2 \to 0$, then $\bar{X} \to \mu$ in probability.**

This problem is asking for a proof of the weak Law of Large Numbers given independent random variables that are NOT identically distributed. However, the proof can use the additional assumption that the variance of the mean converges in probability to zero.

$$E[\bar{X}] = \frac{1}{n}\sum_{i=1}^{n} E[X_i] = \frac{1}{n}\sum_{i=1}^{n}\mu = \mu$$

$$Var(\bar{X}) = \frac{1}{n^2}\sum_{i=1}^{n} Var(X_i) = \frac{1}{n^2}\sum_{i=1}^{n}\sigma_i^2$$

We are given that $n^{-2}\sum_{i=1}^{n}\sigma_i^2 = Var(\bar{X}) \to_p 0$.

Given the convergence of $Var(\bar{X})$, $\bar{X} \to_p \mu$ can be shown as follows:

Proof.

$$\begin{aligned}
P(|\bar{X} - \mu| > \varepsilon) &\leq \frac{Var(\bar{X})}{\varepsilon^2} && \textit{Chebyshev's inequality}\\
\lim_{n\to\infty} P(|\bar{X} - \mu| > \varepsilon) &\leq \lim_{n\to\infty}\frac{Var(\bar{X})}{\varepsilon^2} && \textit{limit}\\
\lim_{n\to\infty} P(|\bar{X} - \mu| > \varepsilon) &\leq 0 && \textit{convergence of variance}
\end{aligned}$$

□

Because the last statement is the definition of convergence in probability, $\bar{X} \to_p \mu$.

6. **(Optional - Based on Rice, Chapter 5, Problem 5) Using moment-generating functions, show that as $n \to \infty, p \to 0$, and $np \to \lambda$, the binomial distribution with parameters n and p tends to the Poisson distribution.** By the Continuity Theorem, can prove convergence in distribution via convergence in mgfs. So, it is necessary to show that the mgf of the binomal converges to the mgf of the Poisson, $e^{\lambda(e^t-1)}$.

The mgf for the binomial RV X is:

$$M_X(t) = ((1-p) + pe^t)^n$$

We need to show that as $n \to \infty, p \to 0$, and $np \to \lambda$, $M_X(t) \to e^{\lambda(e^t-1)}$. This can be demonstrated as follows:

Proof.

$$\begin{aligned}
\lim_{n\to\infty, p\to 0, np\to\lambda} M_X(t) &= \lim_{n\to\infty, p\to 0, np\to\lambda}((1-p) + pe^t)^n && \text{by defn}\\
&= \lim_{n\to\infty, p\to 0, np\to\lambda}(1 - \frac{np}{n} + \frac{npe^t}{n})^n && \textit{algebraic trickery}\\
&= \lim_{n\to\infty, p\to 0, np\to\lambda}(1 + \frac{np(e^t-1)}{n})^n && \textit{combine terms}\\
&= \lim_{n\to\infty}(1 + \frac{\lambda(e^t-1)}{n})^n && \textit{limit of np}\\
&= e^{\lambda(e^t-1)} && \textit{, defn of } e^a
\end{aligned}$$

□

The binomial mgfs therefore converge to a Poisson mgf and, by the Continuity Theorem, the binomial distribution converges to a Poisson distribution.

7. **(Based on Rice, Chapter 5, Problem 16) Suppose that $X_1, ..., X_{20}$ are independent random variables with density functions $f(x) = 3x^2, 0 \leq x \leq 1$. Let $S = X_1 + ... + X_{20}$.**

   (a) **Use the central limit theorem to approximate $P(S \leq 14)$.**

   Grading: 2 pts, full credit if they get the correct answer, 1 points if the approach looks generally valid but answer is wrong

   According to the CLT, for the sum of independent random variables, S_n:

$$\lim_{n\to\infty} P(\frac{S_n - n\mu}{\sigma\sqrt{n}} \leq x) = \Phi(x)$$

   In this case, we are given the pdf of the X_i. The expectation of each X can be found as:

$$\begin{aligned} E[X_n] &= \int_0^1 xf(x)dx \\ &= \int_0^1 3x^3 dx \\ &= [3x^4/4]_0^1 \\ &= 3/4 \end{aligned}$$

   The variance of each X can be found as:

$$\begin{aligned} Var(X_n) &= E[X_n^2] - E[X_n]^2 \\ &= \int_0^1 3x^4 dx - E[X_n]^2 \\ &= 3/5 - 9/16 \\ &= 0.0375 \end{aligned}$$

   The standard deviation of each X can be found as:

$$\begin{aligned} \sigma_{X_n} &= \sqrt{Var(X_n)} \\ &= \sqrt{0.0375} \\ &= 0.1937 \end{aligned}$$

   Given these, $P(S \leq 14)$ can be computed as follows:

$$\begin{aligned} P(S \leq 15) &= P((S - 20 * 3/4)/(0.1937 * \sqrt{20}) \leq (14 - 20 * 3/4)/(0.1937 * \sqrt{20})) \\ &\approx \Phi(-1.154) \\ &= 0.1242 \end{aligned}$$

   In other words, if the expectation on each trial is 3/4, there is only small chance that the sum of 20 such trials will be less than 14.

   (b) **If you are instead asked to approximate $P(S \leq 15)$, what simplifications can be made in the calculation?** For $P(S \leq 15)$, the numerator of the CLT-based probability equation becomes 0, i.e., $15 - 3/4 * 20 = 0$, irrespective of the variance of X_n. So, the probability becomes $\Phi(0)$. Since the normal distribution is symmetric, we know that this probability must be 0.5, i.e., we don't need the implementation of pnorm() in R. So, two simplifications: 1) don't need to compute $Var(X_n)$ and 2) don't need to call pnorm().

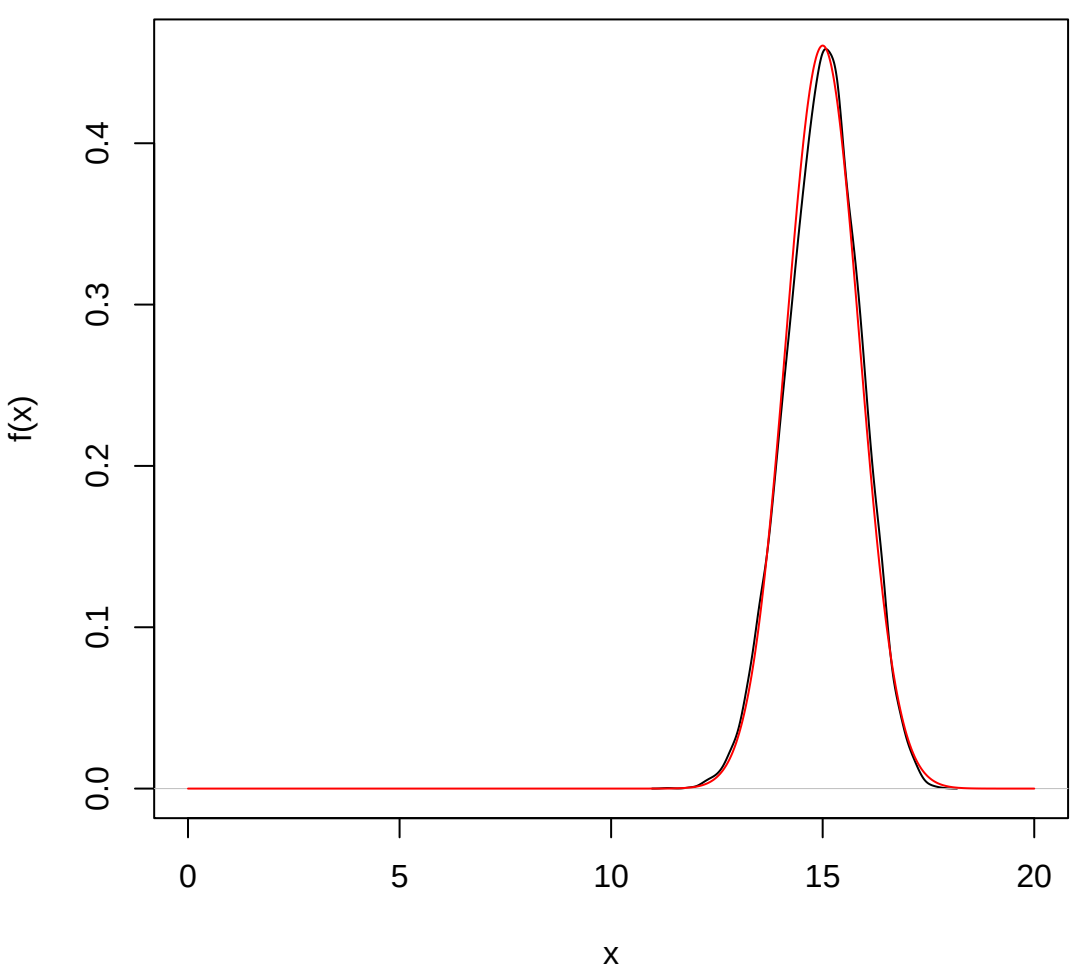
(c) **Validate the approximation by plotting the CLT-based density (compute this using dnorm()) and true density of S. Use the inverse CDF method to simulate from the true density and plot using a kernel density estimate (R code plot(kernel()), we'll learn the details of kernel density estimation later in the course).**

We'll now validate the CLT approximation by simulating from the true density and plotting both the simulated density and the CLT-based approximation ($\mathcal{N}(15, 0.1937^2 * 20)$). To simulate we'll use the inverse CDF method. The CDF of the X_i can be found by integrating the density:

$$\begin{aligned} F(x) &= \int_0^x f(u)du \\ &= \int_0^x 3u^2 du \\ &= \Big[u^3\Big]_0^x \\ &= x^3 \end{aligned}$$

The inverse CDF is therefore $F^{-1}(x) = (p)^{1/3}$. To simulate using the inverse CDF method, we'll generate U(0,1) RVs and then plug into $F^{-1}()$:

```
> n = 10000
> sim.vals = matrix(runif(n*20)^(1/3), nrow=n)
> sum.vals = apply(sim.vals, 1, sum)
> plot(density(sum.vals), xlab="x", ylab="f(x)", xlim=c(0,20))
> x.vals = seq(from=0,to=20, by=0.01)
> points(x.vals, dnorm(x.vals, mean=15, sd=0.1937*sqrt(20)), type="line", col="red")
```



A good match!

8. **(Based on Rice, Chapter 5, Problem 21) We wish to evaluate $I(f) = \int_a^b f(x)dx$ using a numerical estimate. Let g be a density function on $[a, b]$. Generate $X_1, ..., X_n$ from g and estimate I by $\hat{I}(f) = 1/n \sum_{i=1}^n f(X_i)/g(X_i)$.**

   (a) **Show that $E(\hat{I}(f)) = I(f)$**

   Proof.

$$
\begin{aligned}
E[\hat{I}(f)] &= E[1/n \sum_{i=1}^{n} f(X_i)/g(X_i)] \\
&= 1/n \sum_{i=1}^{n} E[f(X_i)/g(X_i)], \textit{ move out constants, E of sum is sum of E} \\
&= E[f(x)/g(x)], \textit{ all expectations are same} \\
&= \int_a^b (f(x)/g(x))g(x)dx, \textit{ definition of expection, wrt g(x)} \\
&= \int_a^b f(x)dx = I(f)
\end{aligned}
$$

□

Note: a similar argument can be used to show that $\hat{I}(f) \to_p I(f)$ by the LLN, we know that $1/n \sum_{i=1}^{n} f(X_i)/g(X_i) \to_p E[f(X)/g(X)]$. This expectation can be computed as follows:

$$\begin{aligned} E[f(X)/g(X)] &= \int_a^b \frac{g(x)f(x)}{g(x)} dx \\ &= \int_a^b f(x) dx \\ &= I(f) \end{aligned}$$

Grading: 2 pts for either approach; 1 pt if they made a reasonable effort.

(b) **Demonstrate the result in a) via simulation with f(x) the density of the standard normal, a=0, b=1 and g(x) the density of the standard uniform distribution. Evaluate for $n = 5, ..., 100$. Plot $\hat{I}(f)$ as a function of n and include a horizontal line at $I(f)$.**

First, let's calculate the integral $I(f)$ using the normal CDF:

$$\begin{aligned} I(f) &= \int_0^1 \phi(x) dx \\ &= \Phi(1) - \Phi(0) \end{aligned}$$

We can compute this using the R pnorm() function:

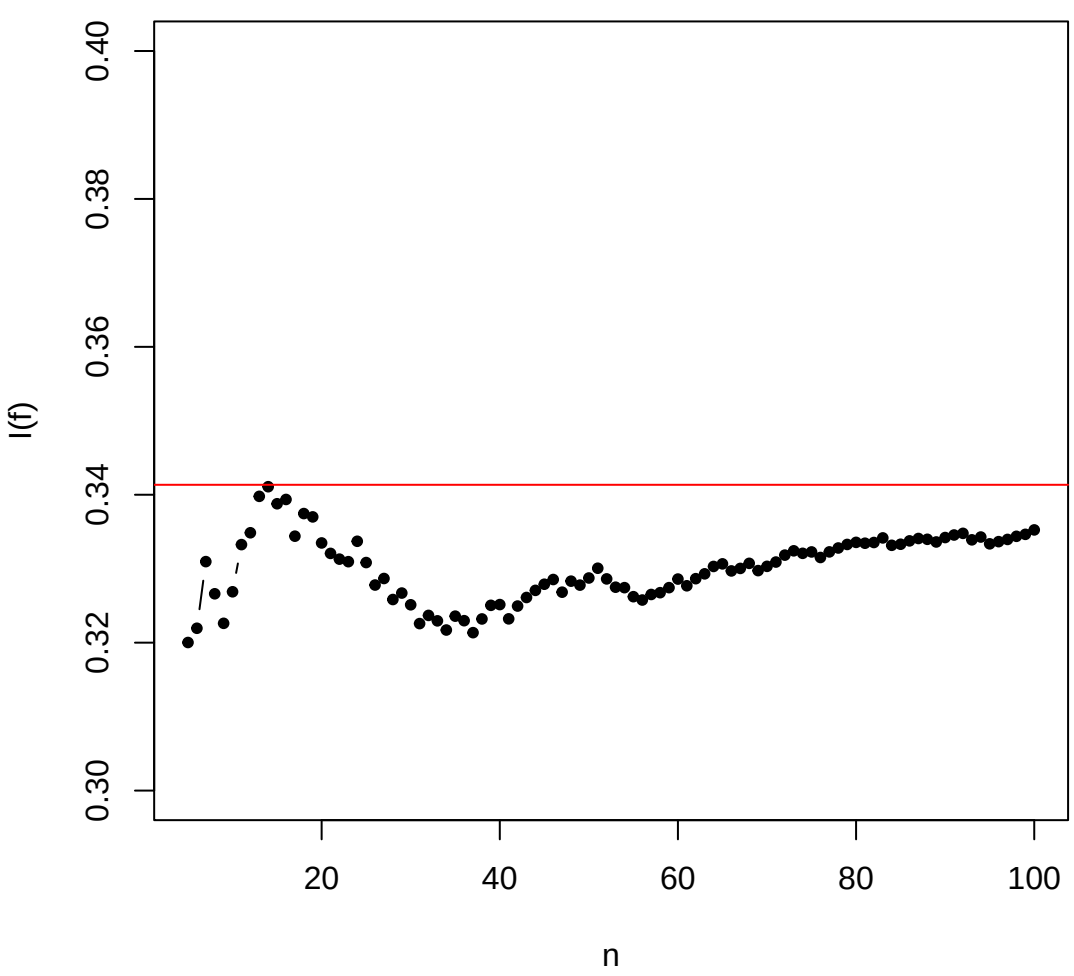
```
> (I_f = pnorm(1) - pnorm(0))
[1] 0.3413447
```

Let's now calculate using the numerical integration approach for n values from 5 to 100:

```
> n.vals = 5:100
> x.vals = runif(100)
> g.vals = rep(1,100) # U(0,1) density is 1 in region
> f.vals = dnorm(x.vals)
> I.est = rep(0, length(n.vals))
> for (i in 1:length(n.vals)) {
+         n = n.vals[i]
+         I.est[i] = mean(f.vals[1:n]/g.vals[1:n])
+ }
> I.est[1:10]
 [1] 0.3200230 0.3219463 0.3309568 0.3266077 0.3226160 0.3268788 0.3332597
 [8] 0.3348603 0.3397845 0.3410872
```

Now plot the estimates vs. n with true I(f) as a horizontal red line:

```
> plot(n.vals, I.est, main="I(f) estimates vs. n", xlab="n", ylab="I(f)", type="b", pch=20,
+         ylim=c(0.3,0.4))
> abline(h=I_f, col="red")
```



As expected, the numerical integration values converge to the true integral (or the R approximation of the value) as n increases.

Grading: 3 pts, full credit if the plotted results look correct; 1.5 pts if the approach looks generally correct but plot looks invalid

(c) **(optional) Can this estimate be improved by choosing g to be other than uniform? Repeat the simulation in b) using a different choice of g (one you think will improve the estimate) and generate a new plot of $\hat{I}(f)$ vs. n that includes the estimates from both g functions. Discuss the relative estimation performance.**

Intuitively, the answer is no. For a general function $f(x)$, $g(x) \sim U()$ will provide the best estimate.

The text gave the hint to compare variances of estimates (i.e., which distribution provides the lowest variance for the estimate). For this, we really need to find the $Var(\hat{I}(f))$ (which is part b). They give the answer in the back of the book and it wasn't assigned but will solve for completeness:

$$
\begin{aligned}
Var(\hat{I}(f)) &= Var(1/n \sum_{i=1}^{n} f(X_i)/g(X_i) \\
&= 1/n^2 \sum_{i=1}^{n} Var(f(X_i)/g(X_i)) \\
&= 1/n Var(f(x)/g(x)) \\
&= 1/n(E[(f(x)/g(x))^2] - E[f(x)/g(x)]^2) \\
&= 1/n(\int_a^b (f(x)/g(x))^2 g(x)dx - (\int_a^b (f(x)/g(x))g(x)dx)^2) \\
&= 1/n(\int_a^b f(x)^2/g(x)dx - (\int_a^b f(x)dx)^2) \\
&= 1/n(\int_a^b f(x)^2/g(x)dx - I(f)^2)
\end{aligned}
$$

Because $g(x)$ is in the denominator of the first term in the variance expression, it is desirable that it has a large density at those places where $f(x)$ also has a large density. To accommodate arbitrary functions $f(x)$, it the uniform distribution is best and provides the lowest overall variance estimator. However, if it were known that a specific type of function was being integrated, it should be better to sample from a probability distribution whose density was aligned with the density of the function.

In this case, we know that the density is $\mathcal{N}(0, 1)$ so sampling from something that has a similar structure will be ideal. One might think to just let $g(x)$ be the density for the standard normal ($\phi(x)$), however, this is not a valid density on $(0, 1)$ (only 34% of the probability mass is in this range). We can make it a valid density by adding to $\phi(x)$ a constant equal to $1 - \int_0^1 \phi(x)dx$ to ensure that the area under $g(x)$ between 0 and 1 is 1.

Let's first define our new g(x) using R:

```
> g_x = function(x) {
+         const = 1-(pnorm(1)-pnorm(0))
+         return (dnorm(x) + const)
+ }
> g_x(0.5)
```

```
[1] 1.010721
```

If we want to simulate from this, we can use the inverse CDF method but that requires that we first compute the CDF:

$$
\begin{aligned}
G(x) &= \int_0^x (c + \phi(x))dx \\
&= cx + \Phi(x) - \Phi(0)
\end{aligned}
$$

Let's implement this in R:

```
> G_x = function(x) {
+         const = 1-(pnorm(1)-pnorm(0))
+         return (const*x + pnorm(x) - pnorm(0))
+ }
> G_x(1)
```

```
[1] 1
```

We'll just brute force the inverse CDF by iteratively checking x vals until we find one that yields a close CDF value:

```
> G_inv = function(p) {
+         x.vals = seq(from=0, to=1,by=0.001)
+         x = 0
+         while(T) {
+                 p.x = G_x(x)
+                 if (p.x >= p) {
+                         return (x)
+                 }
+                 x = x + 0.001
+         }
+ }
> G_inv(0.5)
```

```
[1] 0.48
```

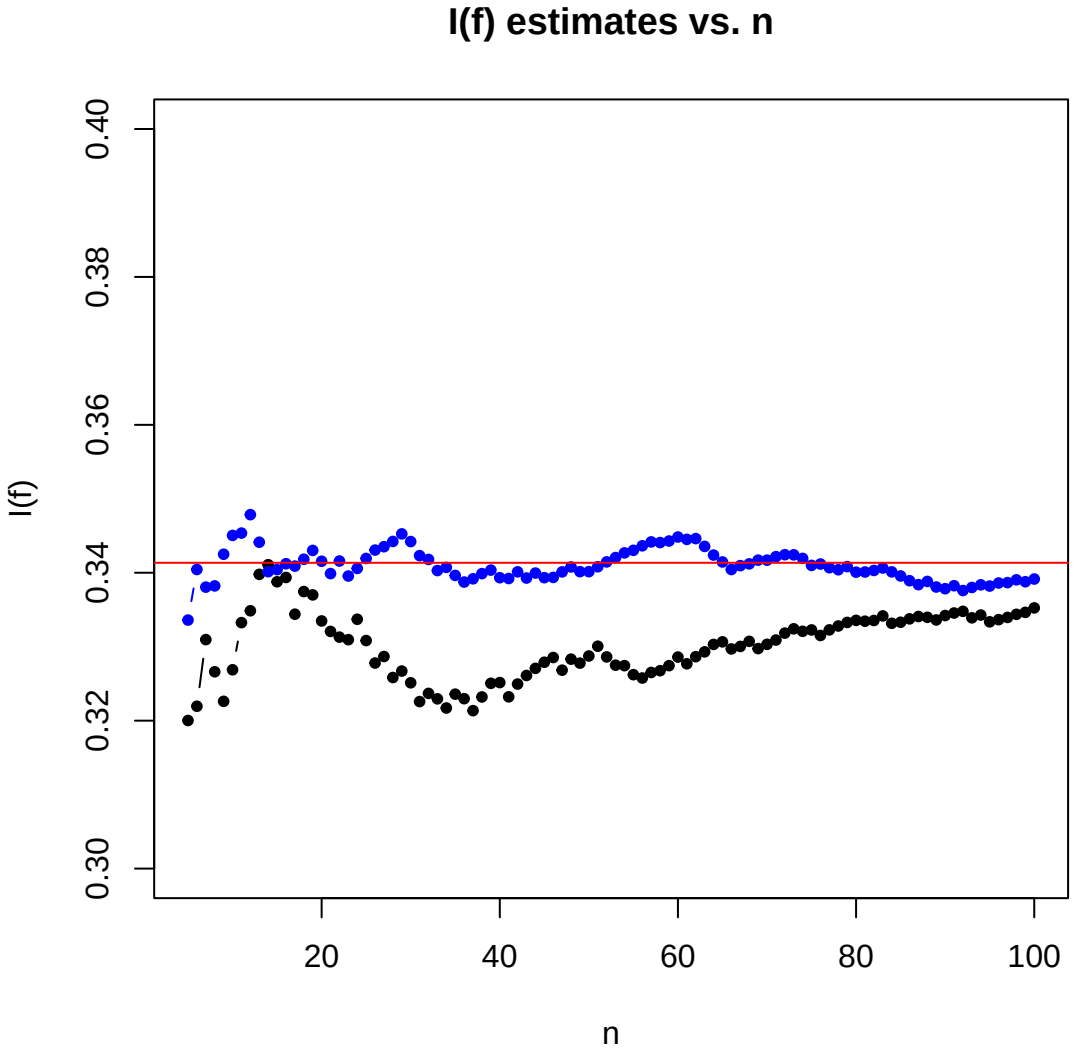
Now recalculate the numerical integral using the new g(x) for n values from 5 to 100:

```
> n.vals = 5:100
> u = runif(100)
> x.vals = sapply(u, G_inv)
> g.vals = g_x(x.vals)
> f.vals = dnorm(x.vals)
> I.est.new = rep(0, length(n.vals))
> for (i in 1:length(n.vals)) {
+         n = n.vals[i]
+         I.est.new[i] = mean(f.vals[1:n]/g.vals[1:n])
+ }
> I.est.new[1:10]
```

```
 [1] 0.3336015 0.3404423 0.3380549 0.3382187 0.3425032 0.3450450 0.3453643
 [8] 0.3478559 0.3441311 0.3401426
```

Now plot the estimates vs. n with true I(f) as a horizontal red line. The estimates generated using g(x)=U(0,1) are in black, the new g(x) is in blue:

```
> plot(n.vals, I.est, main="I(f) estimates vs. n", xlab="n", ylab="I(f)", type="b", pch=20,
+         ylim=c(0.3,0.4))
> points(n.vals, I.est.new, type="b", pch=20, col="blue")
> abline(h=I_f, col="red")
```



Although it is difficult to judge from just this simulation study, the new g(x) that is based on the shape of the standard normal appears to have superior performance.