

# STA3127 Statistical Computing

Jeong Geonwoo

DUE Friday, Nover 10

```
rm(list=ls())
set.seed(2018122062)
```

## Q1.

We wish to generate random numbers  $X \sim \text{Exp}(2)$  and  $Y \sim \text{Exp}(3)$ , where  $\text{Exp}(\lambda)$  is the exponential distribution with rate parameter  $\lambda$ , with some dependence structure between  $X$  and  $Y$  specified by a bivariate Gaussian copula. Generate  $n = 10^5$  pairs of  $(X, Y)$  using the bivariate Gaussian copula with each of the correlation coefficients  $\rho = -0.5$  and  $\rho = 0.5$ . You are only allowed to use the built-in quantile and probability functions of normal distributions (i.e., qnorm and pnorm) with runif for random number generation. You can also use chol function for the Cholesky decomposition if you want.

- i) Let  $(U, V) \sim C_\rho$ , where  $C_\rho$  is the Gaussian copula with correlation parameter  $\rho$ . For each of  $\rho = -0.5$  and  $\rho = 0.5$ , give histograms of the generated  $U$  and  $V$  (to see the marginals) and the scatter plot between the generated  $U$  and  $V$  (to see the joint distribution). Discuss your

First, I will define the functions I use in this problem. I use Cholesky decomposition and Box-Muller Transformation for generating Gaussian copula using only runif

```
##### Q1 #####
## i) ##

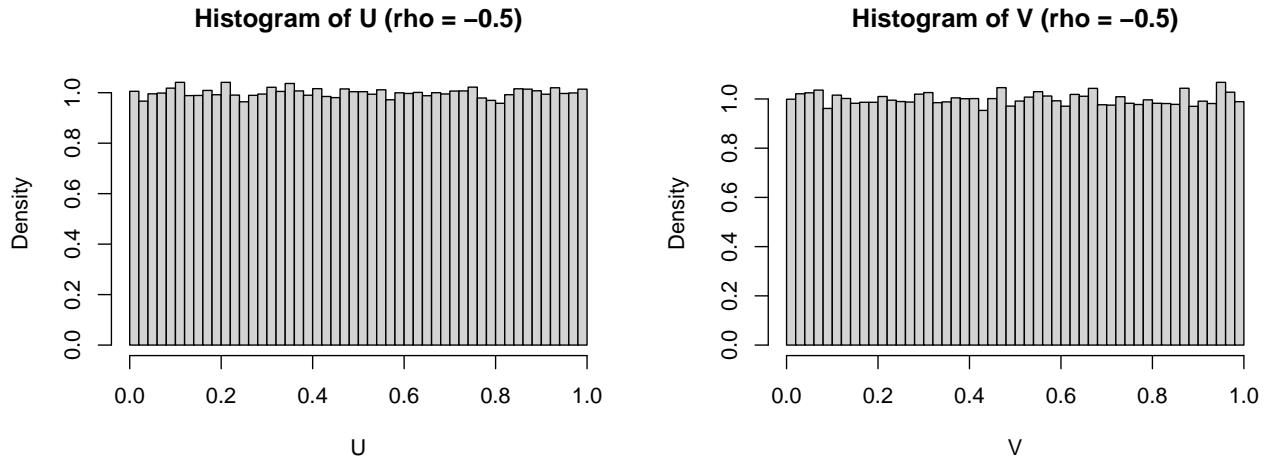
# function 1. Define the function to generate Gaussian distribution
# (using Cholesky decomposition and Box-Muller Transformation)
my_mvrnorm <- function(n, mean, sigma) {
  L <- chol(sigma) # Cholesky decomposition
  my_norm <- function(n) { # Box-Muller Transformation
    u1 <- runif(n/2); u2 <- runif(n/2)
    z1 <- sqrt(-2 * log(u1)) * cos(2 * pi * u2)
    z2 <- sqrt(-2 * log(u1)) * sin(2 * pi * u2)
    return(c(z1, z2))
  }
  # Standard Normal distribution
  z <- matrix(my_norm(n * length(mean)), ncol = length(mean), byrow = TRUE)
  mean + z %*% L # cov matrix
}

# function 2. Define the function to generate Gaussian copula
my_gaussian_copula <- function(n, rho) {
  sigma <- matrix(c(1, rho, rho, 1), nrow=2) # Correlation matrix
  normal_samples <- my_mvrnorm(n, mean=c(0, 0), sigma=sigma) # Generate bivariate normal
  U <- pnorm(normal_samples[,1]); V <- pnorm(normal_samples[,2])
  return(list(U = U, V = V))
}
```

For  $\rho = -0.5$ , I will draw the histogram and scatter plot of  $U, V$

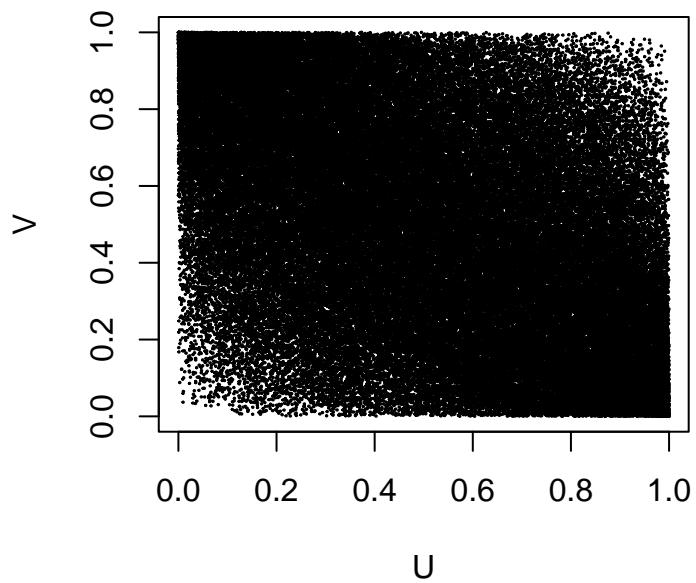
```
# Generate samples for rho = -0.5
gaussian.samples.neg <- my_gaussian_copula(1e5, -0.5)

# Histograms for rho = -0.5
par(mfrow=c(1,2))
hist(gaussian.samples.neg$U, xlab="U", breaks=50, freq=FALSE,
      main="Histogram of U (rho = -0.5)")
hist(gaussian.samples.neg$V, xlab="V", breaks=50, freq=FALSE,
      main="Histogram of V (rho = -0.5)")
```



```
# Scatter plot for rho = -0.5
par(mfrow=c(1,1))
plot(gaussian.samples.neg$U, gaussian.samples.neg$V,
      main="Scatter plot of U vs V (rho = -0.5)",
      xlab="U", ylab="V", pch=19, cex=0.1)
```

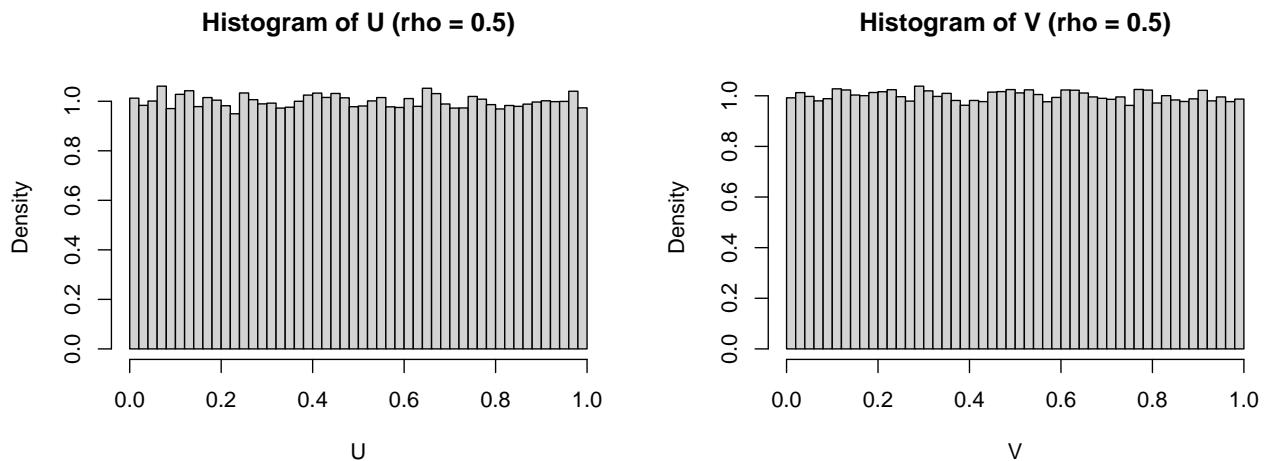
**Scatter plot of U vs V (rho = -0.5)**



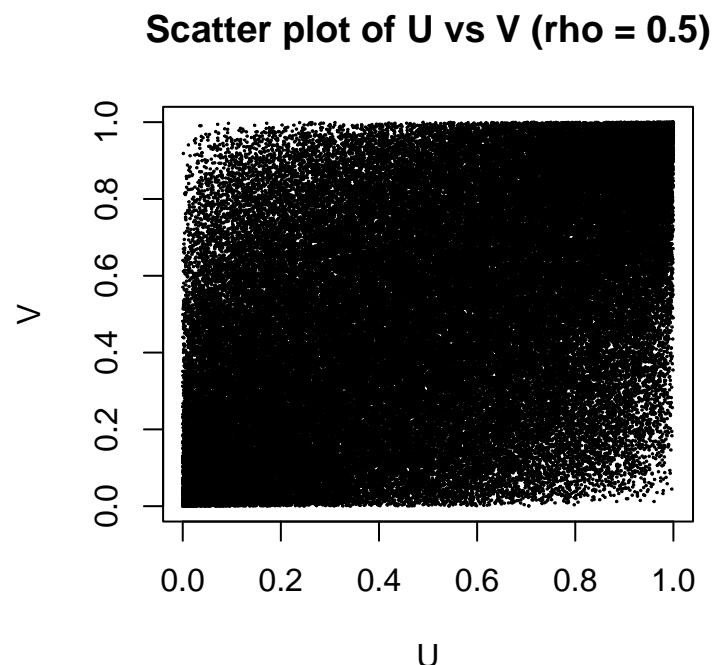
For  $\rho = 0.5$ , I will draw the histogram and scatter plot of  $U, V$

```
# Generate samples for rho = 0.5
gaussian.samples.pos <- my_gaussian_copula(1e5, 0.5)

# Histograms for rho = 0.5
par(mfrow=c(1,2))
hist(gaussian.samples.pos$U, xlab="U", breaks=50, freq=FALSE,
      main="Histogram of U (rho = 0.5)")
hist(gaussian.samples.pos$V, xlab="V", breaks=50, freq=FALSE,
      main="Histogram of V (rho = 0.5)")
```



```
# Scatter plot for rho = 0.5
par(mfrow=c(1,1))
plot(gaussian.samples.pos$U, gaussian.samples.pos$V,
      main="Scatter plot of U vs V (rho = 0.5)",
      xlab="U", ylab="V", pch=19, cex=0.1)
```



From the histograms of U and V, the marginal probability distribution of each variable is uniform on the interval [0,1]. Also from the scatter plot, as the sign of  $\rho$  changes, the correlation between U and V is reversed, and the plot becomes symmetrical. Therefore, we can say that the Gaussian copula is well constructed.

ii) For each of  $\rho = -0.5$  and  $\rho = 0.5$ , give histograms of the generated X and Y (to see the marginals) and the scatter plot between the generated X and Y (to see the joint distribution). Discuss your results.

First, I will define the functions I use in this problem. I used Inverse transformation method.

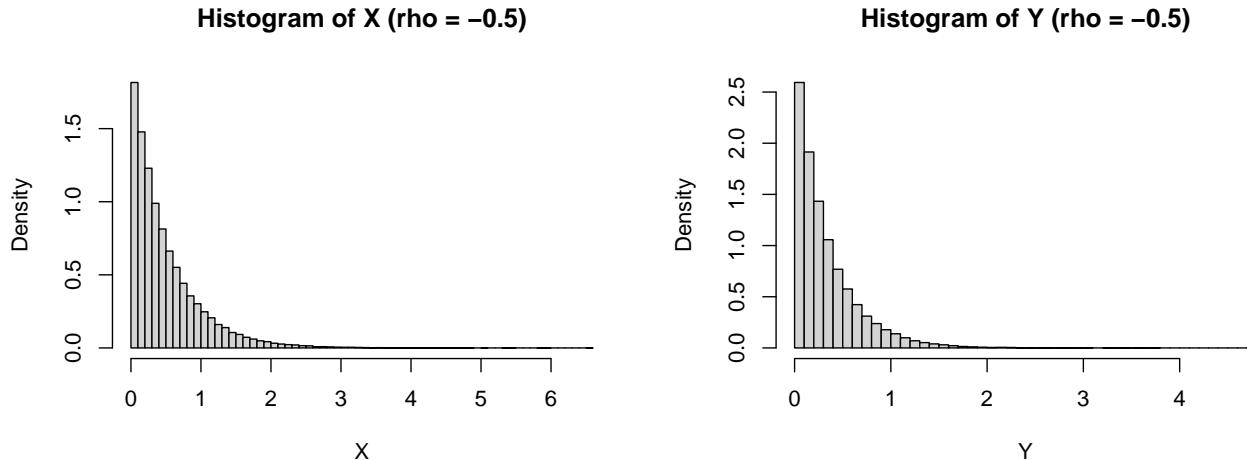
$$\begin{aligned} X &\sim \text{Exp}(\lambda) \\ P(X = x) &= \lambda e^{-\lambda x} \\ P(X \leq x) &= 1 - \lambda e^{-\lambda x} \\ \therefore U &= 1 - \lambda e^{-\lambda x} \\ X &= -\log(1 - U)/\lambda \\ \text{where } U &\sim \text{Unif}(0, 1) \end{aligned}$$

```
# Function to transform
transform_to_exponential <- function(U, V, lambda1, lambda2) {
  X <- -log(1 - U) / lambda1
  Y <- -log(1 - V) / lambda2
  return(list(X = X, Y = Y))
}

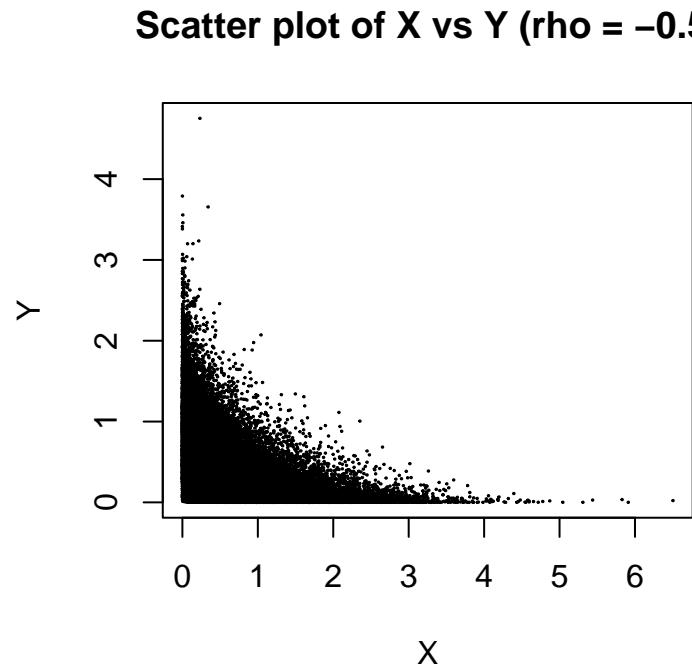
# Transform for both rho values
exp.samples.neg <- transform_to_exponential(gaussian.samples.neg$U,
                                              gaussian.samples.neg$V,
                                              2, 3)
exp.samples.pos <- transform_to_exponential(gaussian.samples.pos$U,
                                              gaussian.samples.pos$V,
                                              2, 3)
```

For  $\rho = -0.5$ , I will draw the histogram and scatter plot of  $X, Y$

```
# Histograms for rho = -0.5
par(mfrow=c(1,2))
hist(exp.samples.neg$X, breaks=50, freq=FALSE,
     main="Histogram of X (rho = -0.5)", xlab="X")
hist(exp.samples.neg$Y, xlab="Y", breaks=50, freq=FALSE,
     main="Histogram of Y (rho = -0.5)")
```

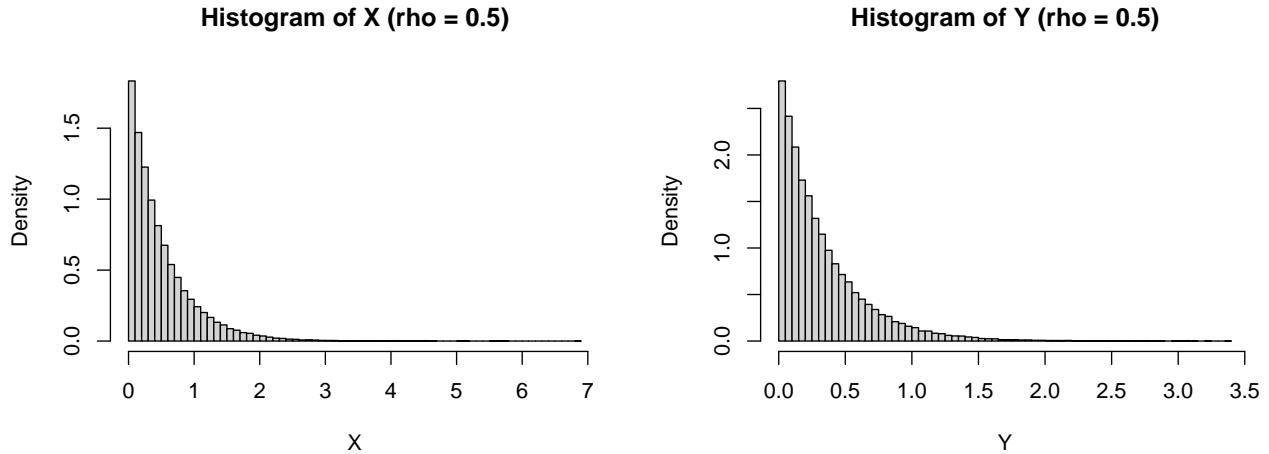


```
# Scatter plot for rho = -0.5
par(mfrow=c(1,1))
plot(exp.samples.neg$X, exp.samples.neg$Y,
     main="Scatter plot of X vs Y (rho = -0.5)",
     xlab="X", ylab="Y", pch=19, cex=0.1)
```

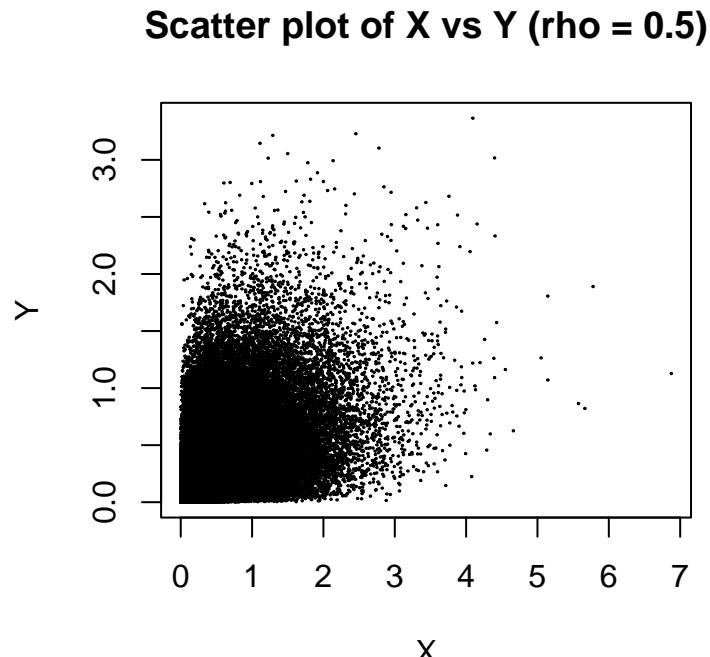


For  $\rho = 0.5$ , I will draw the histogram and scatter plot of  $X, Y$

```
# Histograms for rho = 0.5
par(mfrow=c(1,2))
hist(exp.samples.pos$X, xlab="X", breaks=50, freq=FALSE,
     main="Histogram of X (rho = 0.5)")
hist(exp.samples.pos$Y, xlab="Y", breaks=50, freq=FALSE,
     main="Histogram of Y (rho = 0.5)")
```



```
# Scatter plot for rho = 0.5
par(mfrow=c(1,1))
plot(exp.samples.pos$X, exp.samples.pos$Y,
     main="Scatter plot of X vs Y (rho = 0.5)",
     xlab="X", ylab="Y", pch=19, cex=0.1)
```



From the histograms of  $X$  and  $Y$ , the marginal probability distribution of each variable follow  $\text{Exp}(2)$  and  $\text{Exp}(3)$ , respectively. Also from the scatter plot, as the sign of  $\rho$  changes, the correlation between  $U$  and  $V$  is reversed. Therefore, we can say that th Gaussian copula with exponential marginal distribution is well constructed.

iii) Compare  $\rho$  and the sample correlation coefficient between  $X$  and  $Y$ . Discuss your answer.

```
corr_XY.neg <- cor(exp.samples.neg$X, exp.samples.neg$Y)
corr_XY.pos <- cor(exp.samples.pos$X, exp.samples.pos$Y)
cat("rho : -0.5 ||| sample correlation :", corr_XY.neg, "\n",
    "rho : 0.5 ||| sample correlation :", corr_XY.pos)

## rho : -0.5 ||| sample correlation : -0.365038
## rho : 0.5 ||| sample correlation : 0.4535021
```

iv) Now, instead of exponential distributions for the marginals, suppose we wish to generate  $X \sim N(\mu_1, \sigma_1^2)$  and  $Y \sim N(\mu_2, \sigma_2^2)$  with a dependence structure specified by a bivariate Gaussian copula  $C_\rho$  for a correlation coefficient  $\rho$ . Find the joint distribution of  $(X, Y)$ .

$$\begin{aligned} P(X \leq x, Y \leq y) &= H(x, y) \\ &= P(F(X) \leq F(x), G(Y) \leq G(y)) \\ &= C_{X,Y}(F(x), G(y)) \\ &\text{where } F(X) = \Phi\left(\frac{x - \mu_1}{\sigma_1}\right), G(Y) = \Phi\left(\frac{y - \mu_2}{\sigma_2}\right) \\ &\text{Let } U = \frac{X - \mu_1}{\sigma_1}, V = \frac{Y - \mu_2}{\sigma_2} \end{aligned}$$

Then We can write the density function as following : \ (I will use the variable transformation for expressing following density function using  $X, Y$ )

$$\begin{aligned} C_{X,Y}(F(x), G(y)) &= \int_{-\infty}^{\Phi^{-1}(\Phi(\frac{x - \mu_1}{\sigma_1}))} \int_{-\infty}^{\Phi^{-1}(\Phi(\frac{y - \mu_2}{\sigma_2}))} \frac{1}{2\pi\sqrt{1 - \rho^2}} \exp\left(-\frac{u^2 + v^2 - 2\rho uv}{2(1 - \rho^2)}\right) dudv \\ &= \int_{-\infty}^{\frac{x - \mu_1}{\sigma_1}} \int_{-\infty}^{\frac{y - \mu_2}{\sigma_2}} \frac{1}{2\pi\sqrt{1 - \rho^2}} \exp\left(-\frac{u^2 + v^2 - 2\rho uv}{2(1 - \rho^2)}\right) dudv \end{aligned}$$

From  $U = \frac{X - \mu_1}{\sigma_1}$  and  $V = \frac{Y - \mu_2}{\sigma_2}$ , We can say  $\frac{\partial u}{\partial x} = \frac{1}{\sigma_1}$ ,  $\frac{\partial v}{\partial y} = \frac{1}{\sigma_2}$      $\therefore |J| = \begin{vmatrix} \frac{1}{\sigma_1} & 0 \\ 0 & \frac{1}{\sigma_2} \end{vmatrix} = \frac{1}{\sigma_1\sigma_2}$

Therefore,

$$\begin{aligned} C_{X,Y}(F(x), G(y)) &= \int_{-\infty}^x \int_{-\infty}^y \frac{1}{2\pi\sqrt{1 - \rho^2}} \exp\left(-\frac{\left(\frac{x - \mu_1}{\sigma_1}\right)^2 + \left(\frac{y - \mu_2}{\sigma_2}\right)^2 - 2\rho\left(\frac{x - \mu_1}{\sigma_1}\right)\left(\frac{y - \mu_2}{\sigma_2}\right)}{2(1 - \rho^2)}\right) \frac{1}{\sigma_1\sigma_2} dxdy \\ &= \int_{-\infty}^x \int_{-\infty}^y \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1 - \rho^2}} \exp\left(-\frac{\left(\frac{x - \mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x - \mu_1}{\sigma_1}\right)\left(\frac{y - \mu_2}{\sigma_2}\right) + \left(\frac{y - \mu_2}{\sigma_2}\right)^2}{2(1 - \rho^2)}\right) dxdy \end{aligned}$$

Above result is exactly same to CDF of multivariate gaussian distribution. Therefore, the joint distribution of  $(X, Y)$  is :  $\begin{bmatrix} X \\ Y \end{bmatrix} \sim N\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}\right)$

## Q2.

Suppose that we are interested in the sampling distribution of the sample variance  $S_n^2 = (n - 1)^{-1} \sum_{i=1}^n (X_i - \bar{X})^2$ , where  $X_i \stackrel{\text{iid}}{\sim} \text{Exp}(1)$ . Generate  $n = 10$  observations from  $\text{Exp}(1)$ .

$$\begin{aligned} X &\sim \text{Exp}(\lambda) \\ P(X = x) &= \lambda e^{-\lambda x} \\ P(X \leq x) &= 1 - \lambda e^{-\lambda x} \\ \therefore U &= 1 - \lambda e^{-\lambda x} \\ X &= -\log(1 - U)/\lambda \\ \text{where } U &\sim \text{Unif}(0, 1) \end{aligned}$$

```
n <- 10; r <- 1000
my_rexp <- function(n, lambda){
  return(replicate(n, -log(1-runif(1))/lambda ))
}
exp_10 <- my_rexp(n, 1)
```

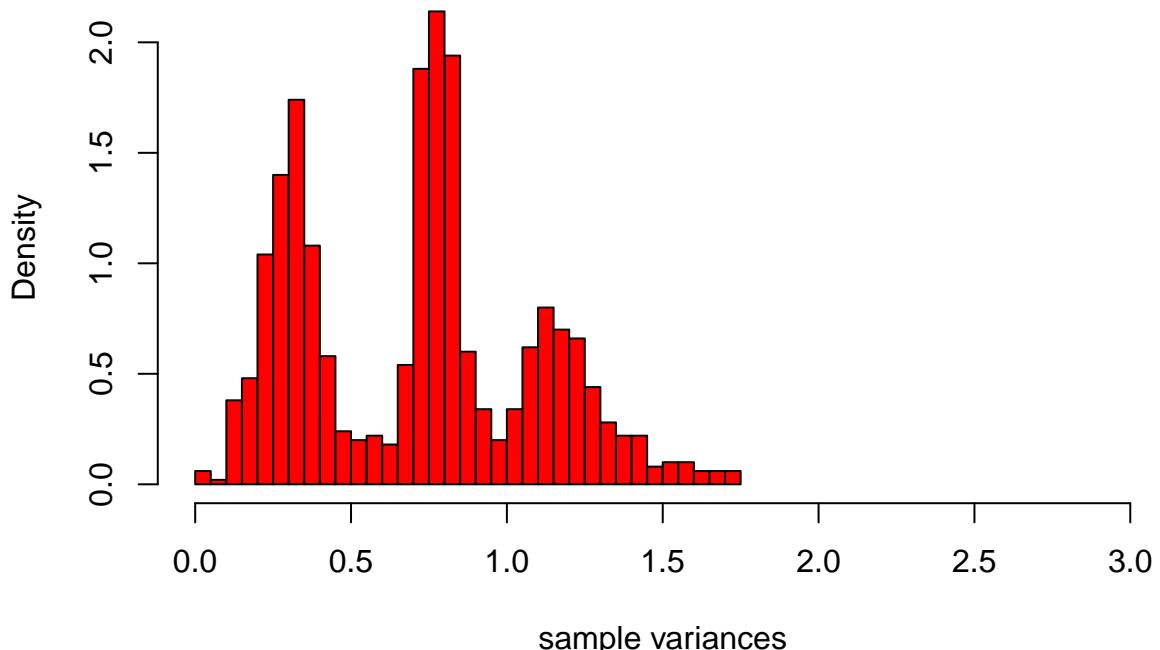
- i) Using the  $n = 10$  observations generated above, obtain the bootstrap sampling distribution of  $S_{10}^2$  with  $r = 1000$  bootstrap samples and draw its histogram.

I will generate  $\text{index matrix} \in \mathbb{R}^{(r,n)}$  and will index the 10 samples using index matrix. If  $(i,j)$  element of index matrix is  $k \in (1, \dots, n)$ , I will get the matrix that have  $(i,j)$  element as  $\exp_{10}[k]$ . Then, I will calculate the variance by row. From that, I will get the  $r$  sample variances where each sample size is  $n$

```
# I will generate (r*n) numbers from {1,2,...,n} and construct "index_matrix" shape : (r,n)
index_matrix <- matrix(floor(n * runif(r * n)) +1, nrow = r, ncol = n)
# I will index exp_10 using "index_matrix"
# then each row is : 10 samples from exp_10 (with replacement)
bootstrap_samples <- matrix(exp_10[index_matrix], nrow = r, ncol = n)
# and get the variance by row.
bootstrap_var <- apply(bootstrap_samples, 1, var)

hist(bootstrap_var, main = "Bootstrap Sample Variances",
     xlab = "sample variances", xlim = range(c(0, 3)), breaks=50, freq=FALSE, col="red")
```

**Bootstrap Sample Variances**



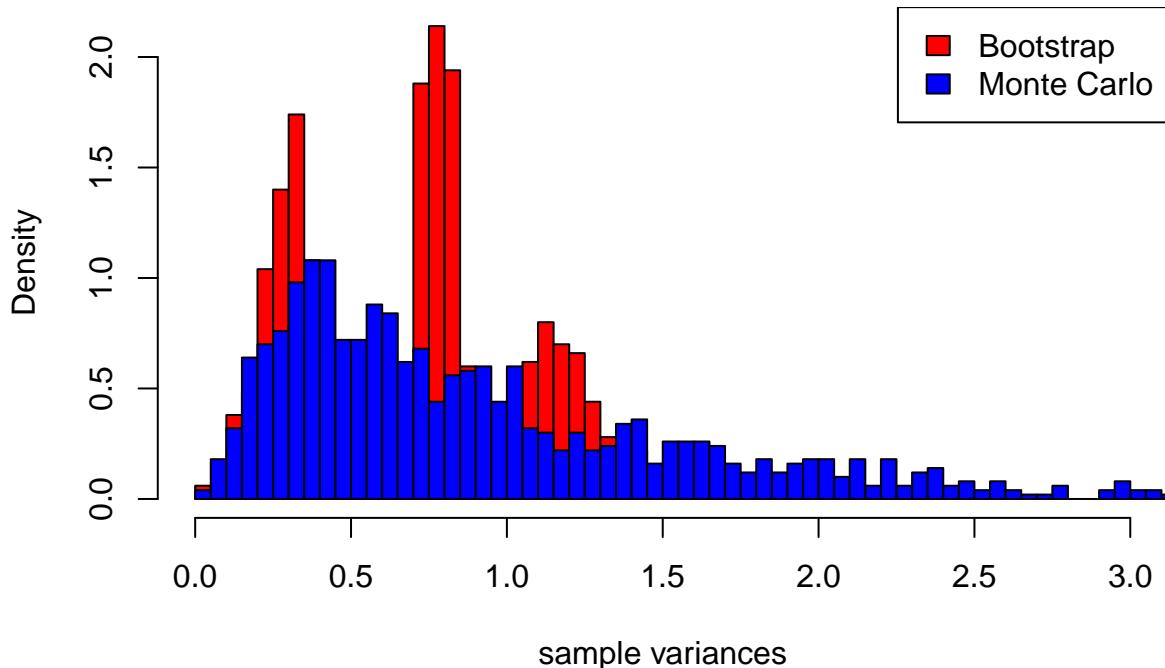
Above figure(red) is a histogram of the sample variances obtained by generating  $n=10$  numbers from  $\exp(1)$  and re-sampling the 10 numbers (with replacement) from given 10 numbers  $r=1000$  times.

ii) By generating additional random draws from  $\text{Exp}(1)$ , obtain a Monte Carlo sampling distribution of  $S_{10}^2$  and overlay its histogram on the one obtained in i).

```
monte_carlo_samples <- matrix(my_rexp(n*r, 1), nrow=r, ncol=n)
monte_carlo_var <- apply(monte_carlo_samples, 1, var)

hist(bootstrap_var, main = "Bootstrap and Monte Carlo Sample Variances",
      xlab = "sample variances", xlim = range(c(0, 3)), breaks=50, freq=FALSE, col="red")
hist(monte_carlo_var, add=TRUE, col="blue", breaks=200, freq=FALSE)
legend("topright", legend = c("Bootstrap", "Monte Carlo"), fill = c("red", "blue"))
```

### Bootstrap and Monte Carlo Sample Variances



Bootstrap sampling distribution of  $S_{10}^2$  (red) obtained in (i) is a histogram of the process of generating 10 samples from  $\text{Exp}(1)$  and re-sampling with replacement from the given 10 samples to obtain the bootstrap sample variance,  $r=1000$  times.

The Monte Carlo sampling distribution of  $S_{10}^2$  (blue) obtained in ii) is a histogram of  $r=1000$  iterations of the process of generating  $n=10$  samples from  $\text{Exp}(1)$  to obtain the sample variance.

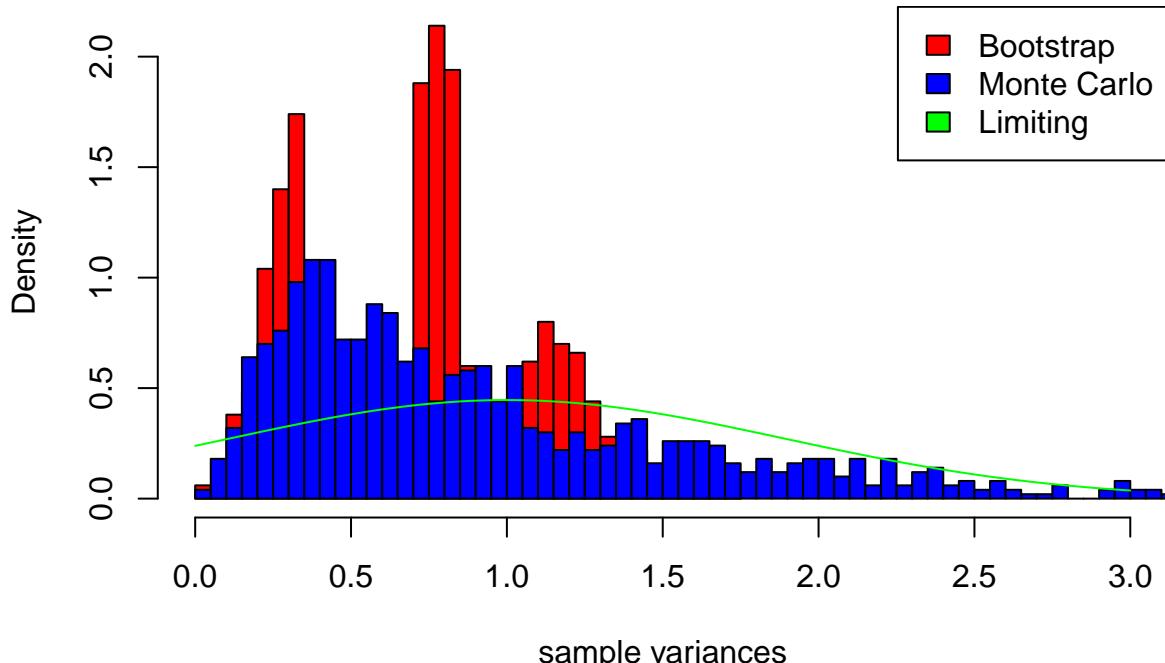
iii) It is well known that, for any data distribution, the sample variance has the limiting distribution as  $\sqrt{n}(S_n^2 - \sigma^2) \xrightarrow{d} N(0, \mu_4 - \sigma^4)$ , where  $\mu_4 = E[(X_i - E(X_i))^4]$  is the fourth central moment and  $\sigma^2$  is the population variance. Using that  $\mu_4 = 9$  and  $\sigma^2 = 1$  for  $\text{Exp}(1)$ , obtain the asymptotic sampling distribution of  $S_{10}^2$  and overlay its density on the plots in i)-ii).

If we know that  $\mu_4 = 9$  and  $\sigma^2 = 1$ , we can directly find the limiting distribution of the sample variance from the facts given in the problem.

$$\begin{aligned}\sqrt{n}(S_n^2 - \sigma^2) &\xrightarrow{d} N(0, \mu_4 - \sigma^4) \\ S_n^2 &\xrightarrow{d} N\left(\sigma^2, \frac{\mu_4 - \sigma^4}{n}\right) \\ S_n^2 &\xrightarrow{d} N\left(1, \frac{8}{10}\right)\end{aligned}$$

```
mu4 <- 9
sigma2 <- 1
hist(bootstrap_var, main = "Bootstrap and Monte Carlo Sample Variances",
      xlab = "sample variances", xlim = range(c(0, 3)), breaks=50, freq=FALSE, col="red")
hist(monte_carlo_var, add=TRUE, breaks=200, freq=FALSE, col="blue")
# green line : limiting sampling distribution using mu4=9, sigma2=1
curve(dnorm(x, mean = sigma2, sd=sqrt(abs(mu4-sigma2^2)/n)), col = "green", add = TRUE)
legend("topright", legend = c("Bootstrap", "Monte Carlo", "Limiting"),
       fill = c("red", "blue", "green"))
```

## Bootstrap and Monte Carlo Sample Variances



The Monte Carlo sampling distribution of  $S_{10}^2$  (blue) is obtained by sampling **new** 10 samples each 1 to r=1000 iteration. So, the range of sample variance is wide. But Bootstrap sampling distribution of  $S_{10}^2$  (red) is obtained by re-sampling from **given** 10 samples (with replacement). So the range of sample variances can not be wide (relative to blue).

iv) As  $\mu_4$  and  $\sigma^2$  are typically unknown, estimate these parameters using reasonable estimators based on the  $n = 10$  observations above. With the estimates of  $\mu_4$  and  $\sigma^2$ , obtain an approximate sampling distribution of  $S_{10}^2$  and overlay its density on the plots in i)-iii).

If we don't know  $\mu_4$  and  $\sigma^2$ , we can estimate  $\mu$  and  $\sigma^2$ . I used the fourth central moment for estimating  $\mu$  and the mean of sample variance  $E(S_{10}^2)$  for estimating  $\sigma^2$

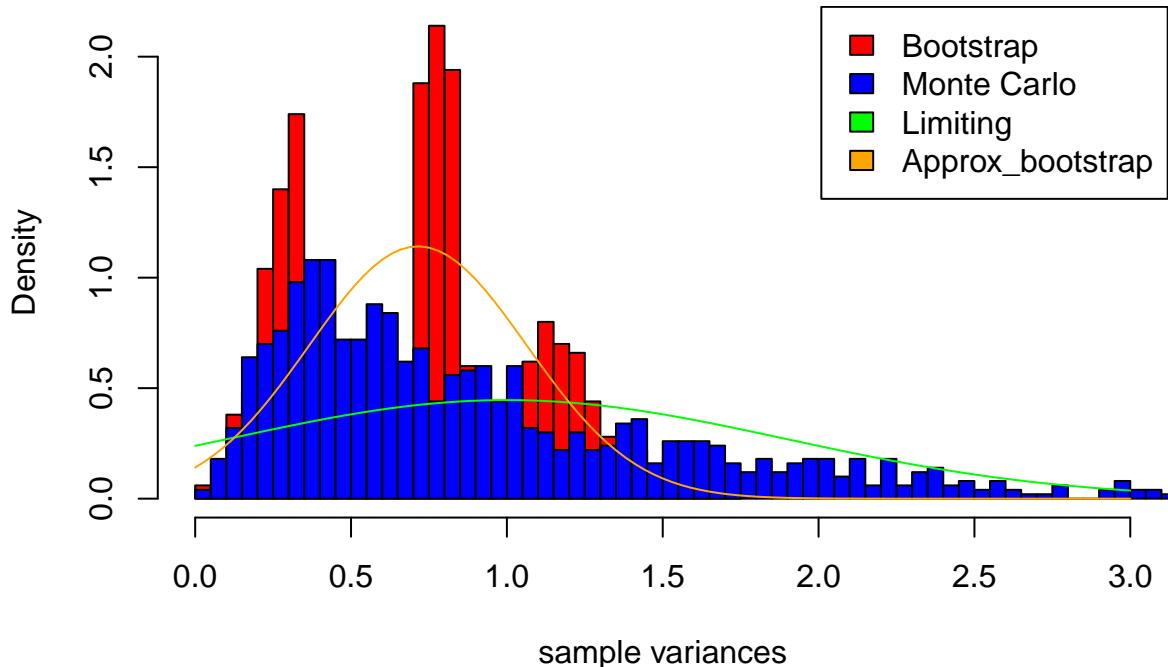
```
hist(bootstrap_var, main = "Bootstrap and Monte Carlo Sample Variances",
     xlab = "sample variances", xlim = range(c(0, 3)), breaks=50, freq=FALSE, col="red")
hist(monte_carlo_var, add=TRUE, col="blue", breaks=200, freq=FALSE)

# green line : limiting sampling distribution using [mu4=9, sigma2=1]
curve(dnorm(x, mean = sigma2, sd=sqrt(abs(mu4-sigma2^2)/n)), col = "green", add = TRUE)

# orange line : approximate sampling distribution using [bootstrap sampling distribution]
mu4 <- mean((bootstrap_samples-mean(bootstrap_samples))^4)
sigma2 <- var(as.vector(bootstrap_samples))
curve(dnorm(x, mean = sigma2, sd=sqrt(abs(mu4-sigma2^2)/n)), col = "orange", add = TRUE)

# legend
legend("topright", legend = c("Bootstrap", "Monte Carlo", "Limiting", "Approx_bootstrap"),
       fill = c("red", "blue", "green", "orange"))
```

## Bootstrap and Monte Carlo Sample Variances



v) Which sampling distributions are similar to and different from each other? Provide valid explanations for the reasons behind these similarities and differences.

First, the approximate sampling distribution of  $S_{10}^2$  (orange) is different from limiting distribution (green). Because we sampled only 10 samples from  $\text{Exp}(1)$  and re-sampled from given 10 samples. Even if we re-sample several times, we can not get sample except given 10 samples. That means, we can not say given 10 samples represent the population (which is  $\text{Exp}(1)$ ). Indeed, the variance of given 10 samples is not identical to known sample variance. Rather, approximate sampling distribution of  $S_{10}^2$  (orange) is similar to the bootstrap sampling distribution of  $S_{10}^2$  (red) because we set  $\mu$  of the approximate sampling distribution (orange) to mean of bootstrap sampling distribution  $E(S_{10}^2)$ .

```
var(exp_10)
```

```
## [1] 0.7943233
```

Second, the Monte Carlo sampling distribution of  $S_{10}^2$  (blue) is more similar to limiting distribution (green) than the bootstrap sampling distribution (red). This is because the more we repeat sampling new samples, the more we can say that sample is representative of the population. However, the reason the Monte Carlo sampling distribution of  $S_{10}^2$  (blue) and limiting distribution (green) are not the same is that  $n$  is not large enough. So we can expect that if we get more samples (which means increase  $n$ ), the Monte Carlo sampling distribution of  $S_{10}^2$  (blue) will be equal to limiting distribution (green).

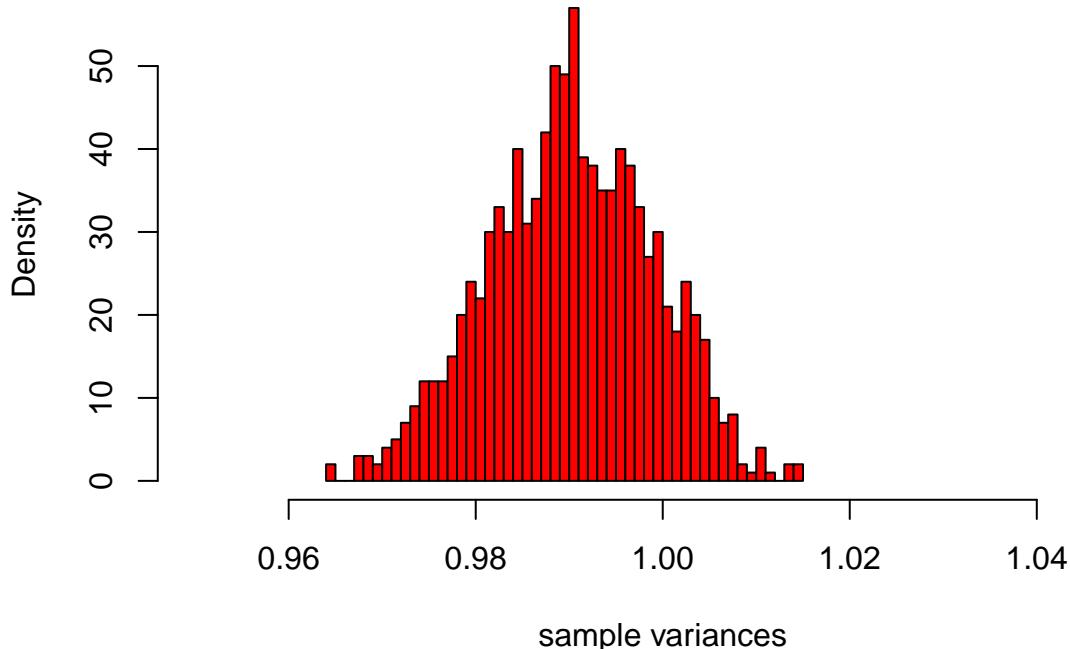
vi) Repeat the above with  $n = 10^5$ .

```
### vi) ####
n <- 1e5; r <- 1000
my_rexp <- function(n, lambda){
  U_matrix <- runif(n)
  return (-log(1-U_matrix)/lambda)
}
exp_1e5 <- my_rexp(n, 1)

### vi-i) ####
index_matrix <- matrix(floor( n * runif(r * n)) +1, nrow = r, ncol = n)
bootstrap_samples <- matrix(exp_1e5[index_matrix], nrow = r, ncol = n)
bootstrap_var <- apply(bootstrap_samples, 1, var)

hist(bootstrap_var, main = "Bootstrap Sample Variances",
      xlab = "sample variances", xlim = range(c(0.95, 1.05)), breaks=50, freq=FALSE, col="red")
```

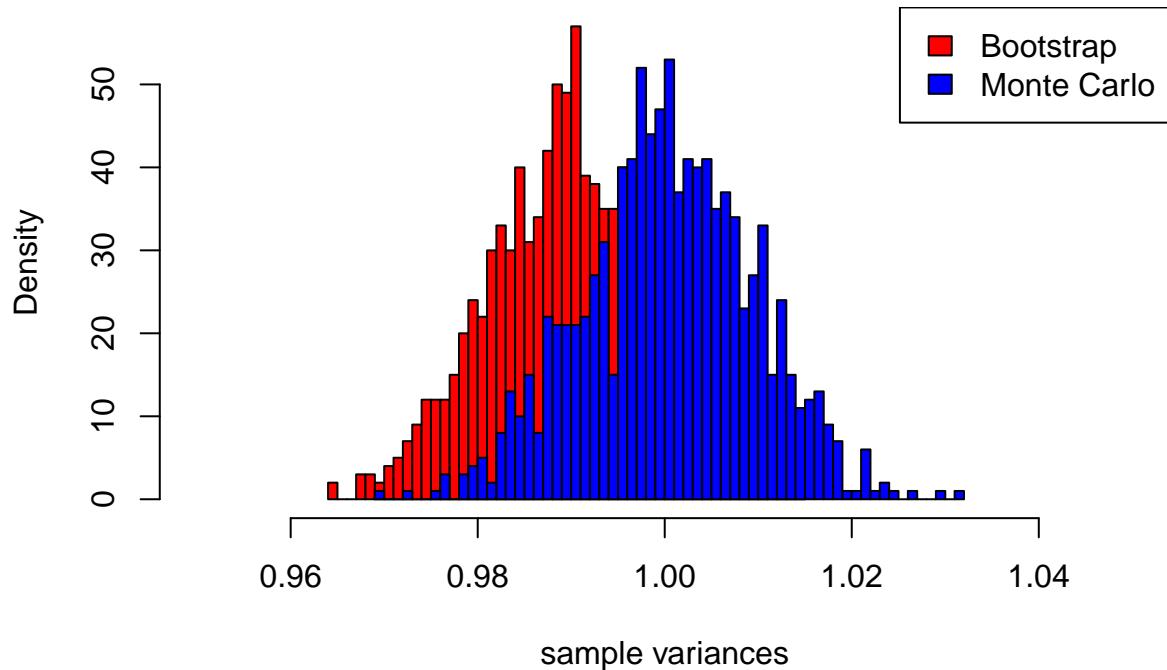
Bootstrap Sample Variances



```
### vi-ii) ####
monte_carlo_samples <- matrix(my_rexp(n*r, 1), nrow=r, ncol=n)
monte_carlo_var <- apply(monte_carlo_samples, 1, var)

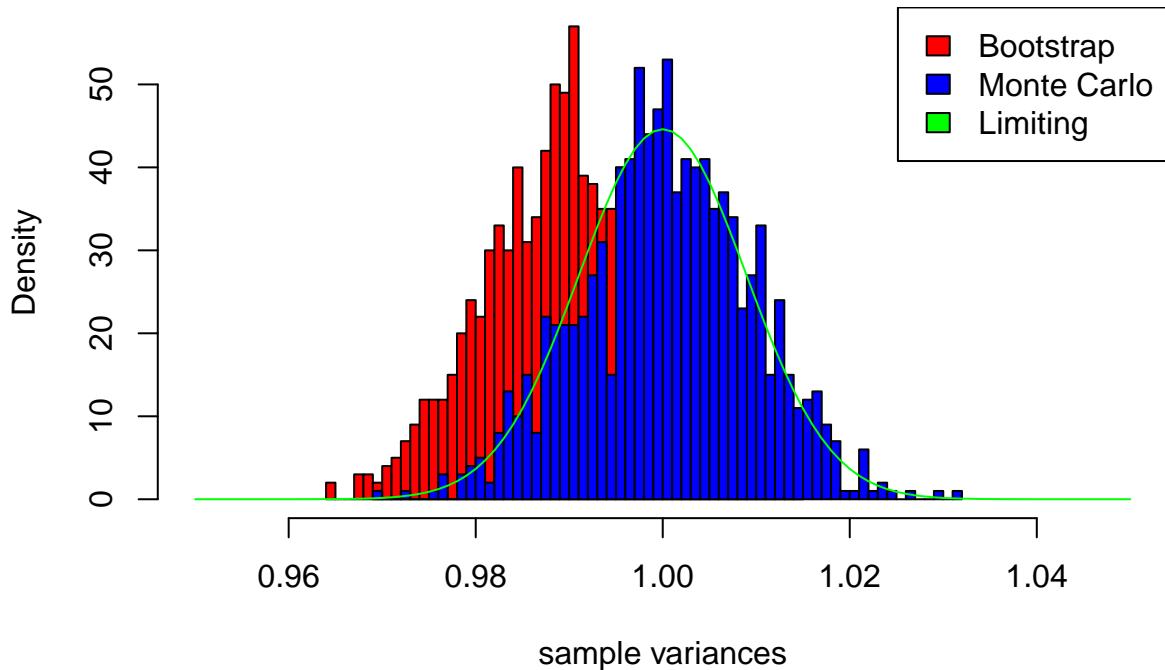
hist(bootstrap_var, main = "Bootstrap and Monte Carlo Sample Variances",
      xlab = "sample variances", xlim = range(c(0.95, 1.05)), breaks=50, freq=FALSE, col="red")
hist(monte_carlo_var, add=TRUE, col="blue", breaks=80, freq=FALSE)
legend("topright", legend = c("Bootstrap", "Monte Carlo"), fill = c("red", "blue"))
```

## Bootstrap and Monte Carlo Sample Variances



```
### vi-iii) ####
mu4 <- 9
sigma2 <- 1
hist(bootstrap_var, main = "Bootstrap and Monte Carlo Sample Variances",
      xlab = "sample variances", xlim = range(c(0.95, 1.05)), breaks=50, freq=FALSE, col="red")
hist(monte_carlo_var, add=TRUE, col="blue", breaks=80, freq=FALSE)
# green line : limiting sampling distribution using mu4=9, sigma2=1
curve(dnorm(x, mean = sigma2, sd=sqrt(abs(mu4-sigma2^2)/n)), col = "green", add = TRUE)
legend("topright", legend = c("Bootstrap", "Monte Carlo","Limiting"),
      fill = c("red", "blue", "green"))
```

## Bootstrap and Monte Carlo Sample Variances



```

### vi-iv) ####
hist(bootstrap_var, main = "Bootstrap and Monte Carlo Sample Variances",
      xlab = "sample variances", xlim = range(c(0.95, 1.05)), breaks=50, freq=FALSE, col="red")
hist(monte_carlo_var, add=TRUE, col="blue", breaks=80, freq=FALSE)

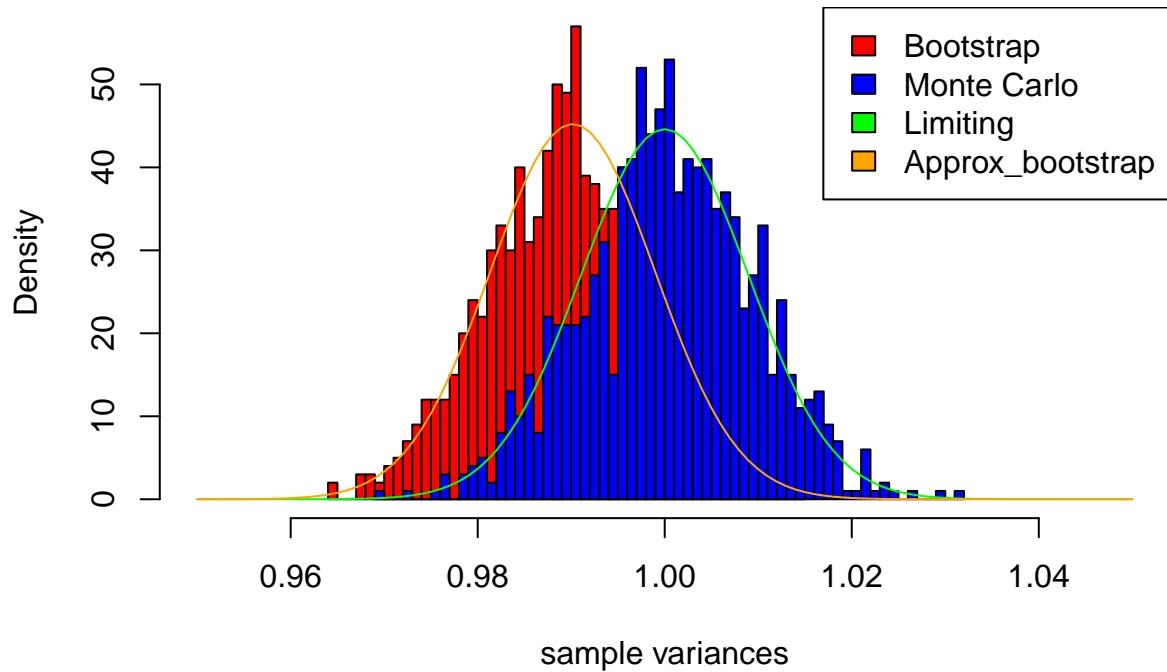
# green line : limiting sampling distribution using [mu4=9, sigma2=1]
curve(dnorm(x, mean = sigma2, sd=sqrt(abs(mu4-sigma2^2)/n)), col = "green", add = TRUE)

# orange line : approximate sampling distribution using [bootstrap sampling distribution]
mu4 <- mean((bootstrap_samples-mean(bootstrap_samples))^-4)
sigma2 <- var(as.vector(bootstrap_samples))
curve(dnorm(x, mean = sigma2, sd=sqrt(abs(mu4-sigma2^2)/n)), col = "orange", add = TRUE)

# legend
legend("topright", legend = c("Bootstrap", "Monte Carlo", "Limiting", "Approx_bootstrap"),
       fill = c("red", "blue", "green", "orange"))

```

## Bootstrap and Monte Carlo Sample Variances



As we expected from v), when  $n = 1e5$ , the Monte Carlo sampling distribution matches the limiting distribution almost exactly. The bootstrap sampling distribution also estimates more accurately than when  $n = 10$ . However, there is nevertheless an error up to the limiting distribution. (compared to the Monte Carlo sampling distribution) This is because the bootstrap sampling distribution used  $n = 1e5$  samples, while the Monte Carlo sampling distribution used  $n \times r = 1e5 \times 1000$  samples. This shows that the bootstrap sampling distribution can make quite accurate estimation with relatively small samples.