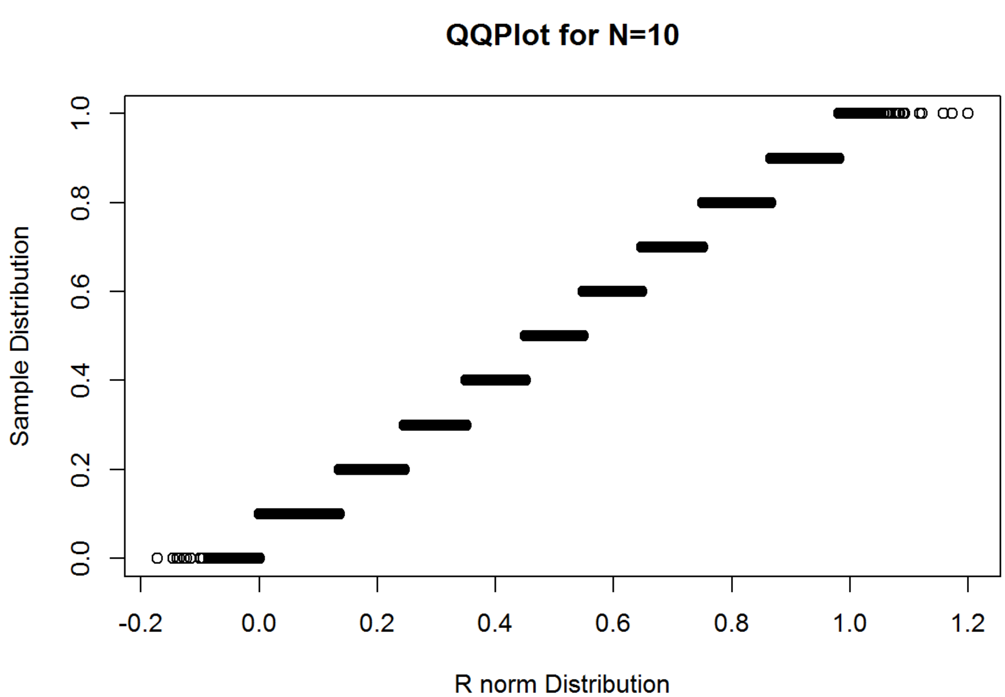
Problem Set 4

Clinton Tepper

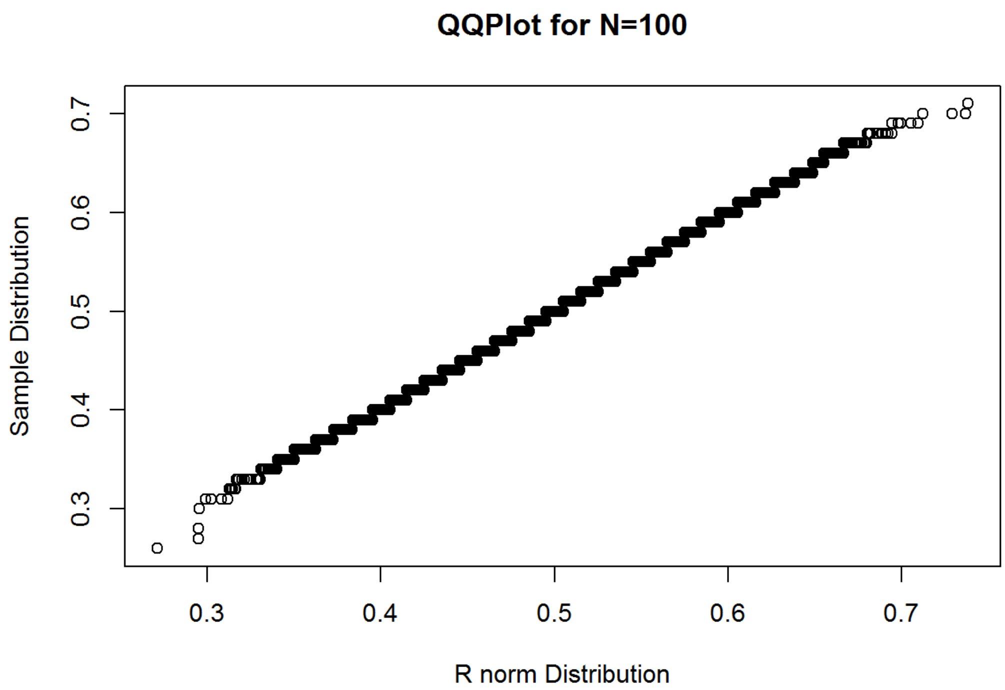
Problem 1

The following block of code creates QQ plots for distributions of averages of ensembles of arbitrary distributions. The plots are against r's random normal distribution (scaled to appropriate parameters. We begin with a Bernoulli distribution with $p = 0.5$.

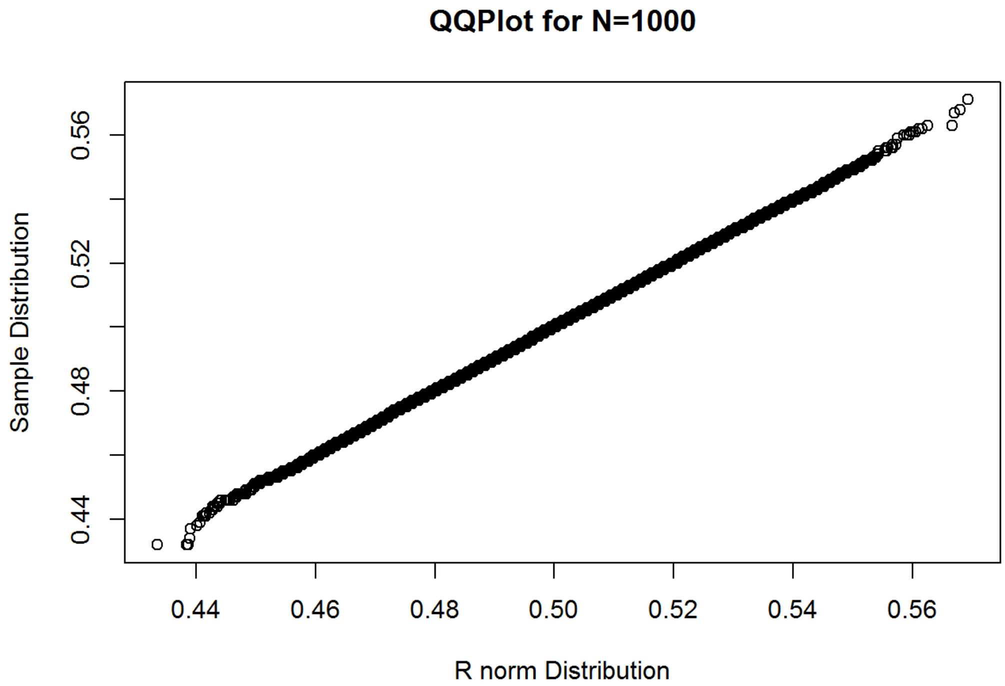
```
#Samples the average of an ensemble of numInstances arbitrary identical distributions
#In: function for sampling random variables (n times),
#number of distributions, number of samples
#Out: a vector of samples, each of which is the average of numInstances distributions
sampleOfAvgs = function(rndFunc, numInstances, samples) {

    return(rep(1, samples) * sapply(rep(1, samples) * numInstances,
    function(n) sum(rndFunc(n)) / numInstances));
    }

#######################Script entry point###########################

#begin with distribution of 0.5
numSamples = 10 ^ 5;
p = .5;

rBern = function(n) rbinom(n, 1, p);
sigma = (p * (1 - p)) ^ .5

nInstances = 10;
qqplot(rnorm(numSamples) * sigma / nInstances ^ .5 + p,
sampleOfAvgs(rBern, nInstances, numSamples),
main = "QQPlot for N=10",
xlab = "R norm Distribution",
ylab = "Sample Distribution");
```



```
nInstances = 100;
qqplot(rnorm(numSamples) * sigma / nInstances ^ .5 + p,
sampleOfAvgs(rBern, nInstances, numSamples),
main = "QQPlot for N=100",
xlab = "R norm Distribution",
ylab = "Sample Distribution");
```

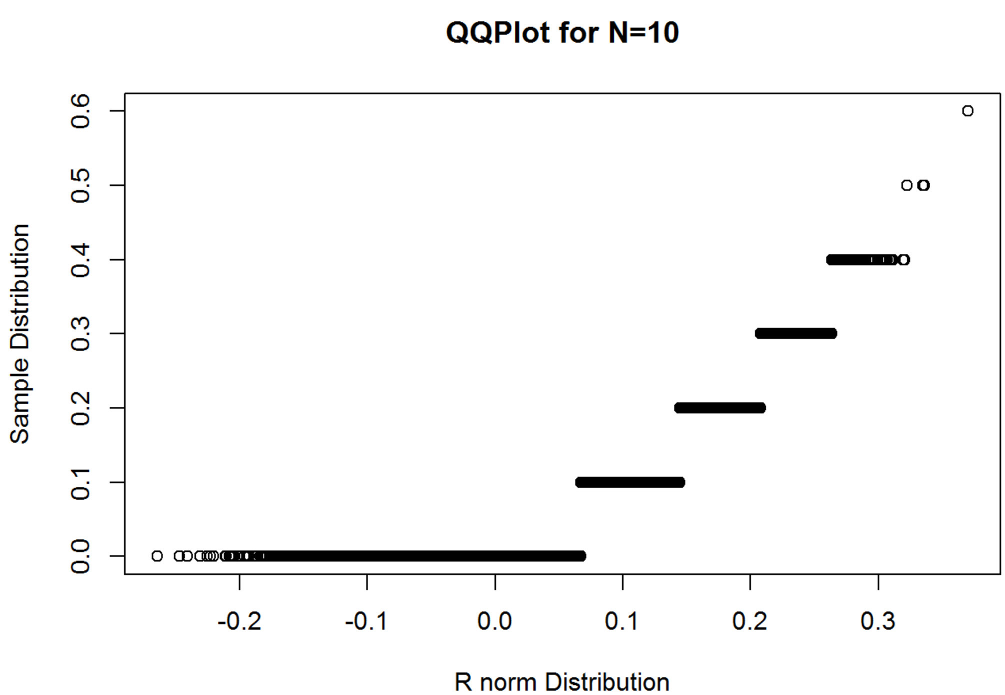



```
nInstances = 1000;
qqplot(rnorm(numSamples) * sigma / nInstances ^ .5 + p,
sampleOfAvgs(rBern, nInstances, numSamples),
main = "QQPlot for N=1000",
xlab = "R norm Distribution",
ylab = "Sample Distribution");
```

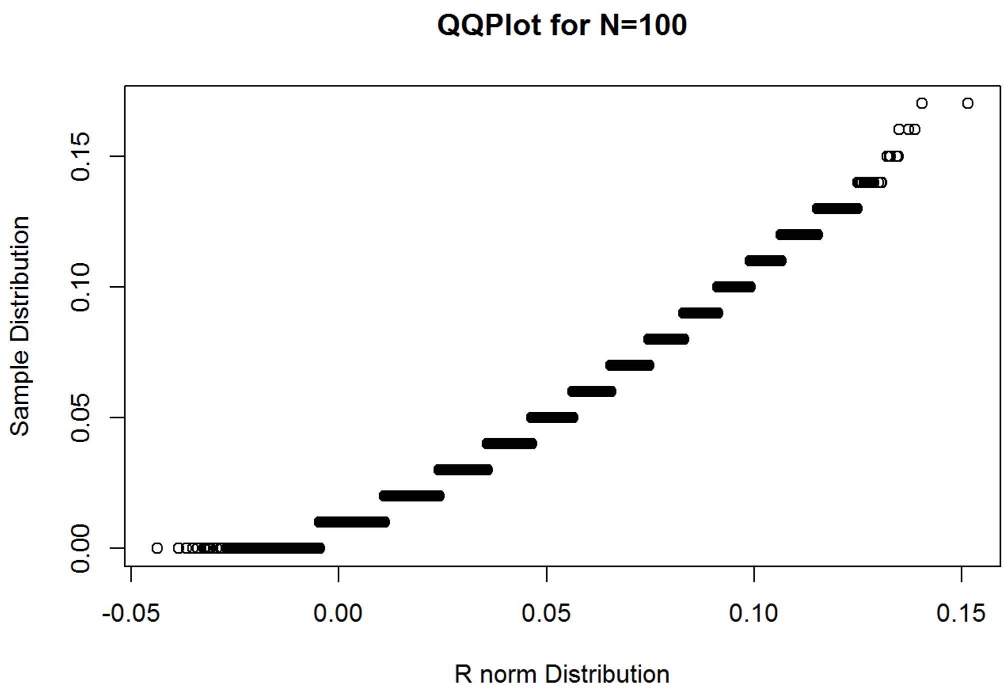


We now repeat the exercise for p=0.05

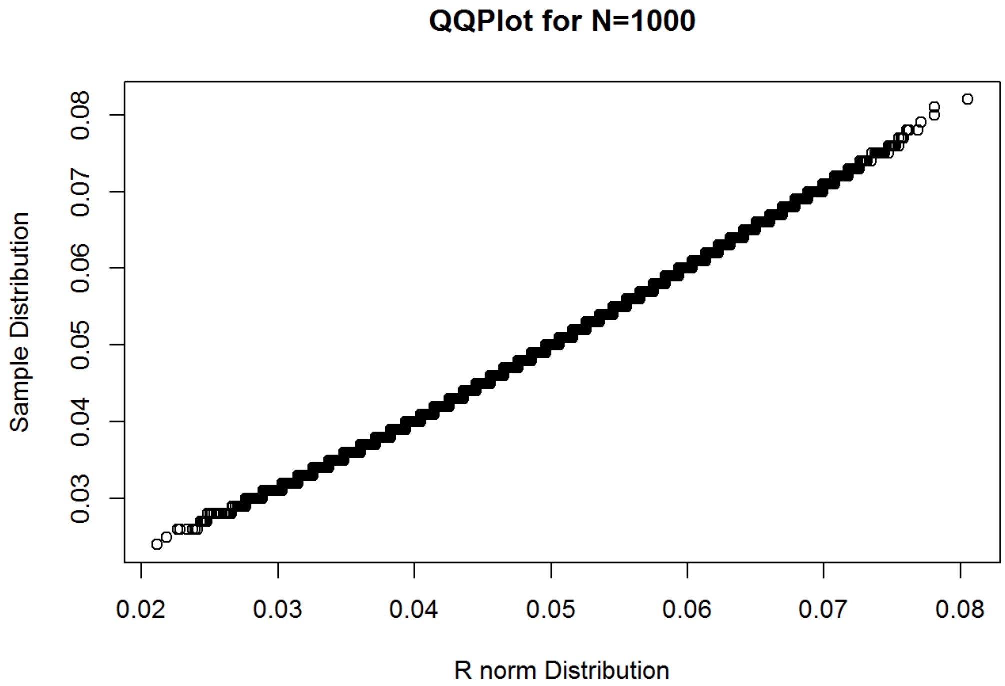
```
p = .05;
rBern = function(n) rbinom(n, 1, p);
sigma = (p * (1 - p)) ^ .5

nInstances = 10;
qqplot(rnorm(numSamples) * sigma / nInstances ^ .5 + p,
sampleOfAvgs(rBern, nInstances, numSamples),
main = "QQPlot for N=10",
xlab = "R norm Distribution",
ylab = "Sample Distribution");
```



```
nInstances = 100;
qqplot(rnorm(numSamples) * sigma / nInstances ^ .5 + p,
sampleOfAvgs(rBern, nInstances, numSamples),
main = "QQPlot for N=100",
xlab = "R norm Distribution",
ylab = "Sample Distribution");
```




```
nInstances = 1000;
qqplot(rnorm(numSamples) * sigma / nInstances ^ .5 + p,
sampleOfAvgs(rBern, nInstances, numSamples),
main = "QQPlot for N=1000",
xlab = "R norm Distribution",
ylab = "Sample Distribution");
```



We conclude by noting that convergence is much faster with the first symmetric bernoulli distribution.

Problem 2

Prove E(S_$n^2) = \sigma^2$

$$
\begin{aligned}
S_n^2 =& \frac{1}{n-1}\sum_{j=1}^{n}\left(X_j-\overline{X}\right)^2\\
E(S_n^2) =& E\left(\frac{1}{n-1}\sum_{j=1}^{n}\left(X_j-\overline{X}\right)^2\right)\\
(n-1)\,E(S_n^2) =& E\left(\sum_{j=1}^{n}\left(X_j^2-2\mu X_j+\overline{X}^2\right)\right)\\
(n-1)\,E(S_n^2) =& E\left(\sum_{j=1}^{n}\left(X_j^2\right)\right)-2E\left(\sum_{j=1}^{n}\left(\overline{X}X_j\right)\right)+E\left(\sum_{j=1}^{n}\overline{X}^2\right)\\
(n-1)\,E(S_n^2) =& E\left(\sum_{j=1}^{n}\left(X_j^2\right)\right)-2E\left(\overline{X}\sum_{j=1}^{n}\left(X_j\right)\right)+nE\left(\overline{X}^2\right)\\
(n-1)\,E(S_n^2) =& E\left(\sum_{j=1}^{n}\left(X_j^2\right)\right)-2nE\left(\overline{X}^2\right)+nE\left(\overline{X}^2\right)\\
(n-1)\,E(S_n^2) =& E\left(\sum_{j=1}^{n}\left(X_j^2\right)\right)-nE\left(\overline{X}^2\right)\\
\frac{(n-1)}{n}E(S_n^2) =& E\left(\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2\right)\right)-E\left(\overline{X}^2\right)\\
\frac{(n-1)}{n}E(S_n^2) =& E\left(\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2\right)\right)-E\left(\left(\frac{1}{n}\sum_{j=1}^{n}X_j\right)^2\right)\\
\frac{(n-1)}{n}E(S_n^2) =& E\left(\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2\right)\right)-\frac{1}{n^2}E\left(\sum_{j=1}^{n}\sum_{k=1}^{n}X_jX_k\right)\\
\frac{(n-1)}{n}E(S_n^2) =& E\left(\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2\right)\right)-\frac{1}{n^2}E\left(\sum_{j=1}^{n}\sum_{k=1}^{n}\left(X_jX_k+\mu^2-X_j\mu-X_k\mu-\left(\mu^2-X_j\mu-X_k\mu\right)\right)\right)\\
\frac{(n-1)}{n}E(S_n^2) =& E\left(\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2\right)\right)-\frac{1}{n}E\left(\frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(X_j-\mu\right)\left(X_k-\mu\right)\right)+\frac{n^2}{n^2}u^2-E\left(\frac{n}{n^2}\sum_{j=1}^{n}X_j\mu\right)-E\left(\frac{n}{n^2}\sum_{k=1}^{n}X_k\mu\right)\\
\text{but }\sigma^2 =& \frac{1}{n}\sum_{j=1}^{n}\sum_{k=1}^{n}\left(X_j-\mu\right)\left(X_k-\mu\right)\\
\therefore\ \frac{(n-1)}{n}E(S_n^2) =& E\left(\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2\right)\right)-\frac{1}{n}\sigma^2+u^2-2E\left(\frac{\mu}{n}\sum_{j=1}^{n}X_j\right)\\
\frac{(n-1)}{n}E(S_n^2) =& E\left(\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2\right)\right)-\frac{1}{n}\sigma^2+u^2-2\mu^2\\
\frac{(n-1)}{n}E(S_n^2) =& E\left(\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2+\mu^2-2X_j\mu-\left(\mu^2-2X_j\mu\right)\right)\right)-\frac{1}{n}\sigma^2-\mu^2\\
\frac{(n-1)}{n}E(S_n^2) =& E\left(\frac{1}{n}\sum_{j=1}^{n}\left(X_j-\mu\right)^2\right)-\frac{1}{n}n\mu^2+\frac{1}{n}E\left(\sum_{j=1}^{n}2X_j\mu\right)-\frac{1}{n}\sigma^2-\mu^2\\
\frac{(n-1)}{n}E(S_n^2) =& E\left(\frac{1}{n}\sum_{j=1}^{n}\left(X_j-\mu\right)^2\right)+2\mu E\left(\frac{1}{n}\sum_{j=1}^{n}X_j\right)-\frac{1}{n}\sigma^2-2\mu^2\\
\frac{(n-1)}{n}E(S_n^2) =& \sigma^2+2\mu^2-\frac{1}{n}\sigma^2-2\mu^2\\
\frac{(n-1)}{n}E(S_n^2) =& \frac{n-1}{n}\sigma^2\\
E(S_n^2) =& \sigma^2\checkmark
\end{aligned}
$$

Prove $S_n^2 \rightarrow^p \sigma^2$

First note that

$$
\begin{aligned}
S_n^2 =& \frac{1}{n-1}\sum_{j=1}^{n}\left(X_j-\overline{X}\right)^2\\
S_n^2 =& \frac{1}{n-1}\sum_{j=1}^{n}\left(X_j^2-2X_j\overline{X}+\overline{X}^2\right)\\
S_n^2 =& \frac{1}{n-1}\sum_{j=1}^{n}\left(X_j^2\right)-\frac{2}{n-1}\sum_{j=1}^{n}X_j\overline{X}+\frac{n}{n-1}\overline{X}^2\\
S_n^2 =& \frac{1}{n-1}\sum_{j=1}^{n}\left(X_j^2\right)-\frac{2n\overline{X}}{n-1}\sum_{j=1}^{n}\frac{1}{n}X_j+\frac{n}{n-1}\overline{X}^2\\
S_n^2 =& \frac{n}{n-1}\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2\right)-\frac{n}{n-1}\overline{X}^2
\end{aligned}
$$

Check this against the probability convergence formula:

$$
P\left(\left|\frac{n}{n-1}\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2\right)-\frac{n}{n-1}\overline{X}^2-\sigma^2\right|>\epsilon\right)\rightarrow 0
$$

By the weak law of large numbers:

$$
\begin{aligned}
\frac{1}{n}\sum_{j=1}^{n}\left(X_j^2\right)\rightarrow^p \sigma^2+\mu^2\\
\overline{X}\rightarrow^p \mu
\end{aligned}
$$

Because $X_n \rightarrow^p X$ and $Y_n \rightarrow^p Y$ together imply $X_nY_n \rightarrow^p XY$, and all other terms are constants:

$$

$$
\begin{aligned}
P\left(\left|\frac{n}{n-1}\sigma^2+\mu^2-\frac{n\mu^2}{n-1}-\sigma^2\right|>\epsilon\right)\rightarrow 0\\
P\left(\left|\sigma^2+\mu^2-\mu^2-\sigma^2\right|>\epsilon\right)\rightarrow 0\\
P\left(0>\epsilon\right)\rightarrow 0\checkmark
\end{aligned}
$$

$$

Problem 3

A

First note that $\overline{X}_1 \rightarrow^p \mu_1$ and $\overline{X}_2 \rightarrow^p \mu_2$. Because $g(X_n) \rightarrow^p g\left(X\right)$ we can calculate $\frac{1}{\bar{X}_2} \rightarrow^p \frac{1}{\mu_2}$. Moreover, because $X_nY_n \rightarrow^p XY$, the derived variable $Y_n \rightarrow^p \frac{\mu_1}{\mu_2}$. ###B We will apply the multivariate delta method as layed out in Wasserman Example 5.16. Theorem 5.15 states that given a sequence of random vectors such that $\sqrt{n}\left(Y_n-\mu\right)\rightsquigarrow N\left(0,\Sigma\right)$. Let g denote $\mathfrak{R}^k \rightarrow \mathfrak{R}$. Then $\sqrt{n}\left(g\left(Y_n\right)-g\left(\mu\right)\right)\rightarrow N\left(0,\left(\nabla' g\right)|_{\mu}\Sigma\left(\nabla g\right)|_{\mu}\right)$

$$
\text{Let: } Y_n = \frac{\overline{X}_1}{\overline{X}_2},\quad g\left(s_1,\ s_2\right)=\frac{s_1}{s_2}
$$

By the CLT:

$$

$$
\sqrt{n}\begin{bmatrix}\overline{X}_1-\mu_1\\ \overline{X}_2-\mu_2\end{bmatrix}\rightsquigarrow N\left(0,\ \Sigma\right)
$$

$$

Note that

$$\nabla g(s) = \begin{bmatrix} \frac{1}{s_2} \\ -\frac{s_1}{s_2^2} \end{bmatrix}$$

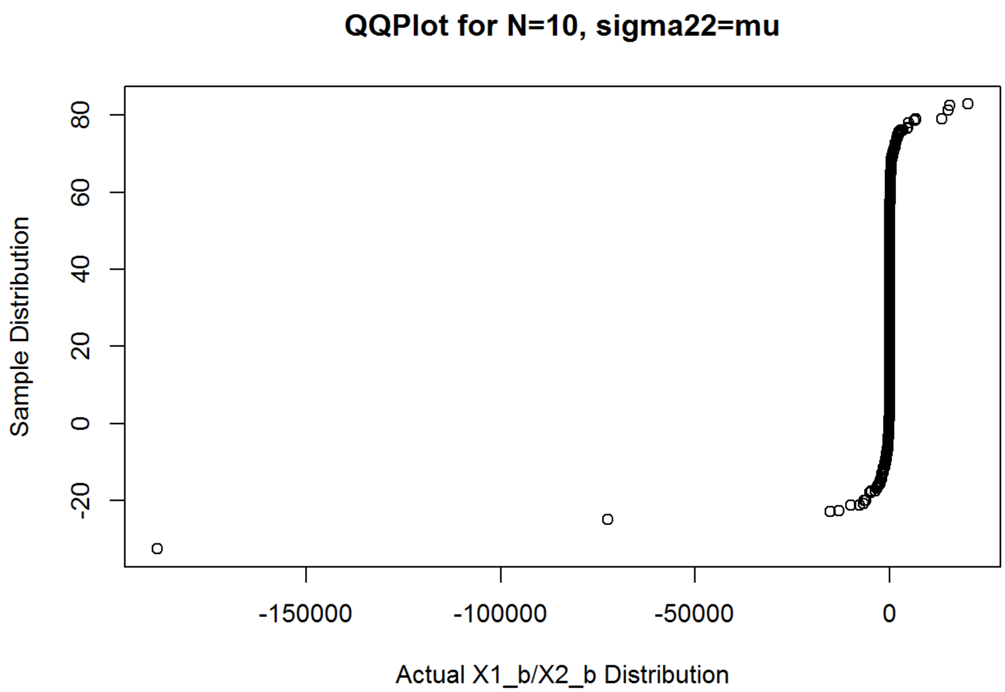
Therefore

$$\sqrt{n}\left(\frac{\overline{X}_1}{\overline{X}_2} - \frac{\mu_1}{\mu_2}\right) \rightsquigarrow N\left(0, \begin{bmatrix} \frac{1}{\mu_2} & -\frac{\mu_1}{\mu_2^2} \end{bmatrix} \Sigma \begin{bmatrix} \frac{1}{\mu_2} \\ -\frac{\mu_1}{\mu_2^2} \end{bmatrix}\right)$$

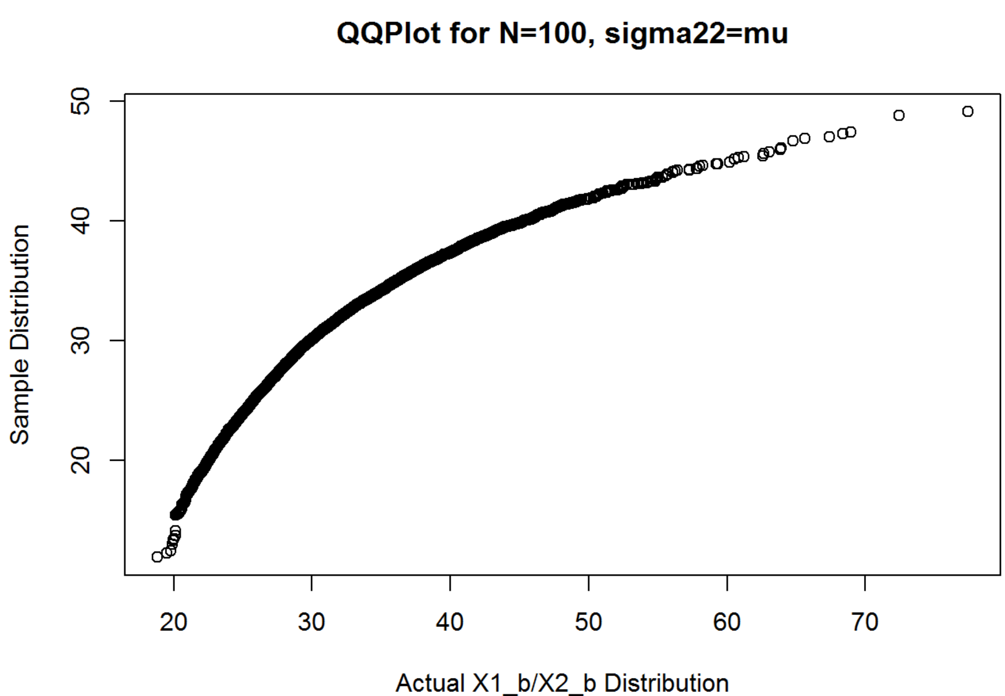
C

This program creates QQ plots for the transformation $\overline{X}_1$ and $\overline{X}_2$ where $\overline{X}_1$ and $\overline{X}_2$ are the sample averages of a bivariate normal. The draws are plotted against the asymptotic approximation, as described above. We begin by outlining the helper methods and generating the plots for the situation where mu_2 equals $sigma_2 2$.

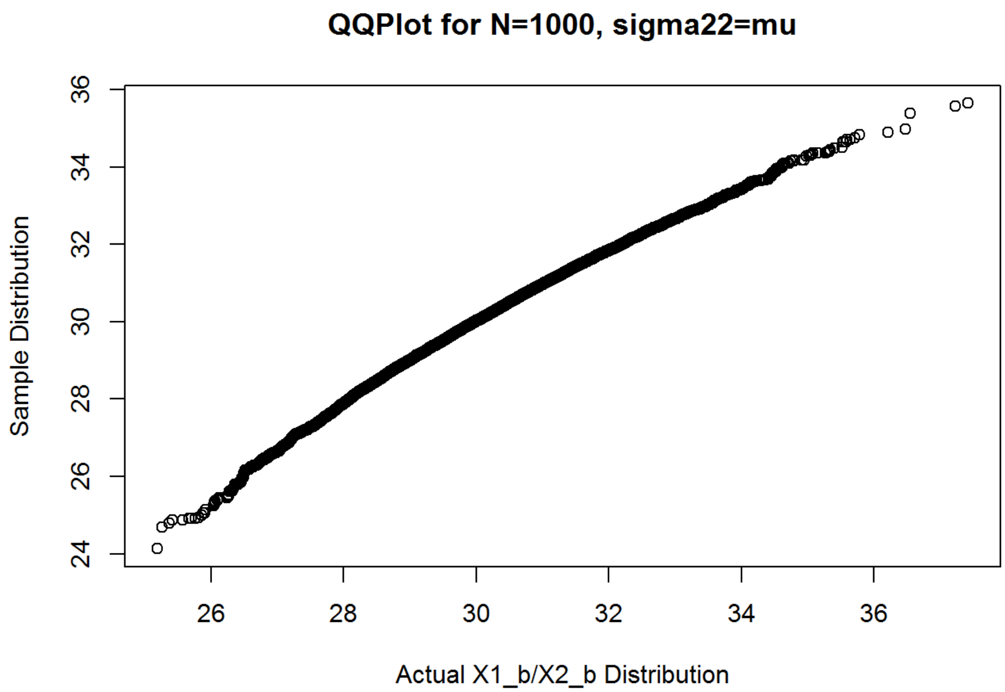
```
bivNorm = function(numSamples, mu, Ut) {
    l = length(mu)
    return(matrix(sapply(1:numSamples, function(x) mu + Ut %*% rnorm(l)),
         numSamples,
         l,
         byrow = TRUE));
}

#Returns an R^2 -> R transformation of the sample average of bivariate normals
#In: transform function g, number of draws, sequence number, mu (vector), cov (matrix)
#Out: a vector of samples of the transformation
getBivTransDraws = function(g, numSamples, n, mu, cov) {
    tCholMat = t(chol(cov));
    bivDraws = matrix(sapply(rep(n, numSamples), function(x) colSums(bivNorm(x, muVec, tCholMat))),
    numDraws, length(muVec), byrow = TRUE);
    return(g(bivDraws[, 1], bivDraws[, 2]));
}

########################Script entry point############################
numDraws = 10 ^ 4;
covMat = matrix(c(1, 1 / 2, 1 / 2, 1 / 3), 2, 2);
# Note- Hilbert matrices are positive definite.
muVec = c(10, 1 / 3);
tranVec = matrix(c(1 / muVec[2], - muVec[1] / (muVec[2] ^ 2)), 2, 1);
sigmaAsympt = sqrt(t(tranVec) %*% covMat %*% tranVec);

seqN = 10;
asymptDraws = rnorm(numDraws) * sigmaAsympt / sqrt(seqN) + muVec[1] / muVec[2];
actDraws = getBivTransDraws(function(x, y) x / y, numDraws, seqN, muVec, covMat);
qqplot(actDraws,
asymptDraws,
main = "QQPlot for N=10, sigma22=mu",
xlab = "Actual X1_b/X2_b Distribution",
ylab = "Sample Distribution");
```




```
seqN = 100;
asymptDraws = rnorm(numDraws) * sigmaAsympt / sqrt(seqN) + muVec[1] / muVec[2];
actDraws = getBivTransDraws(function(x, y) x / y, numDraws, seqN, muVec, covMat);
qqplot(actDraws,
asymptDraws,
main = "QQPlot for N=100, sigma22=mu",
xlab = "Actual X1_b/X2_b Distribution",
ylab = "Sample Distribution");
```

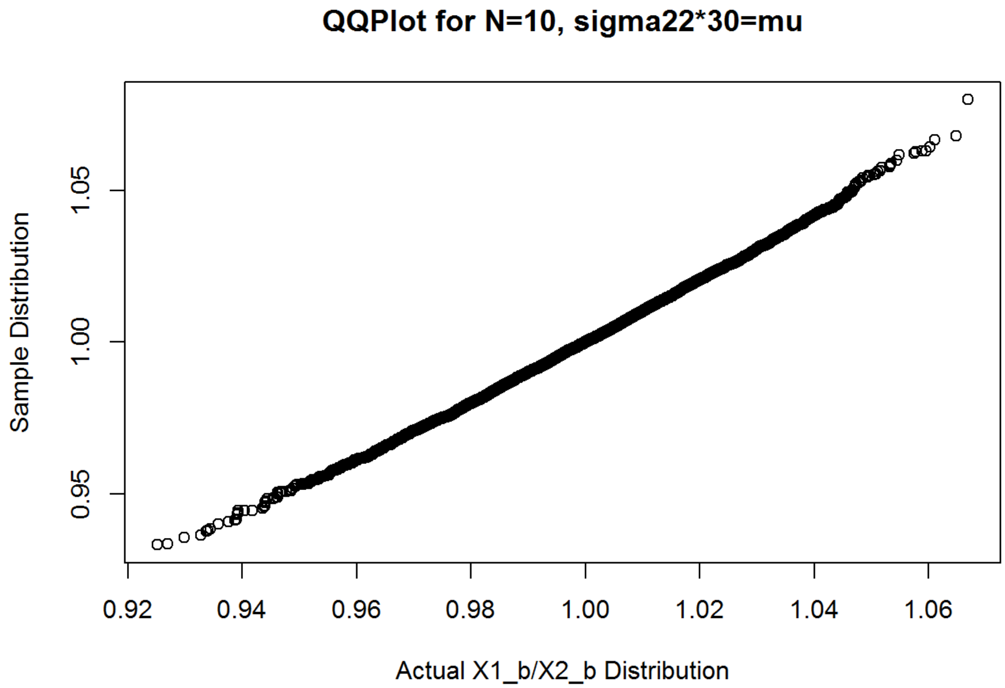


```
seqN = 1000;
asymptDraws = rnorm(numDraws) * sigmaAsympt / sqrt(seqN) + muVec[1] / muVec[2];
actDraws = getBivTransDraws(function(x, y) x / y, numDraws, seqN, muVec, covMat);
qqplot(actDraws,
asymptDraws,
main = "QQPlot for N=1000, sigma22=mu",
xlab = "Actual X1_b/X2_b Distribution",
ylab = "Sample Distribution");
```

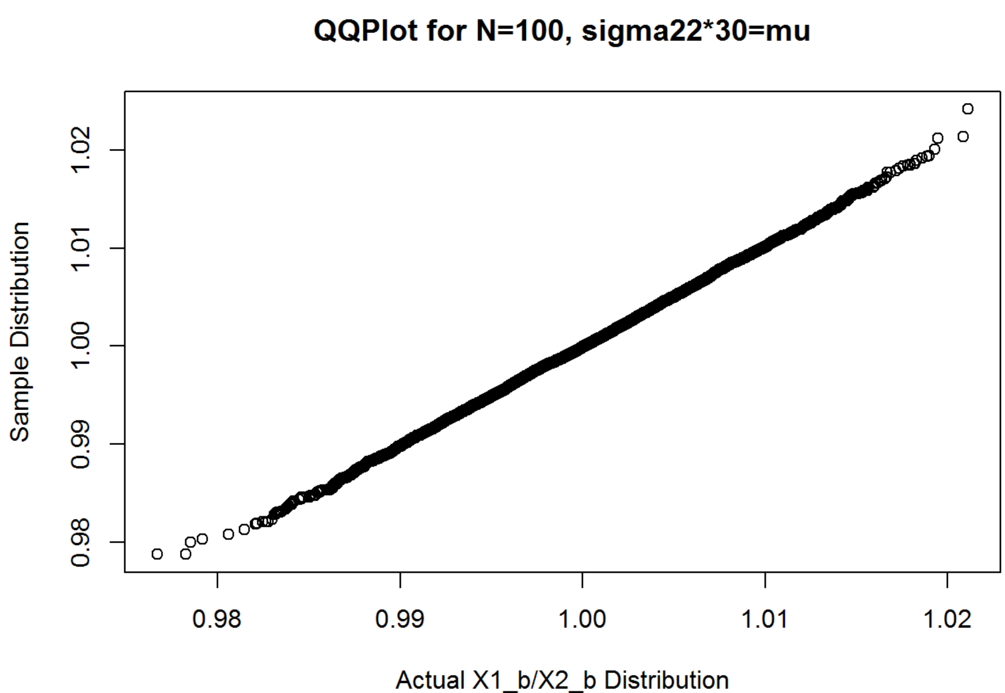


We repeat the exercise with a mu that is different than sigma.

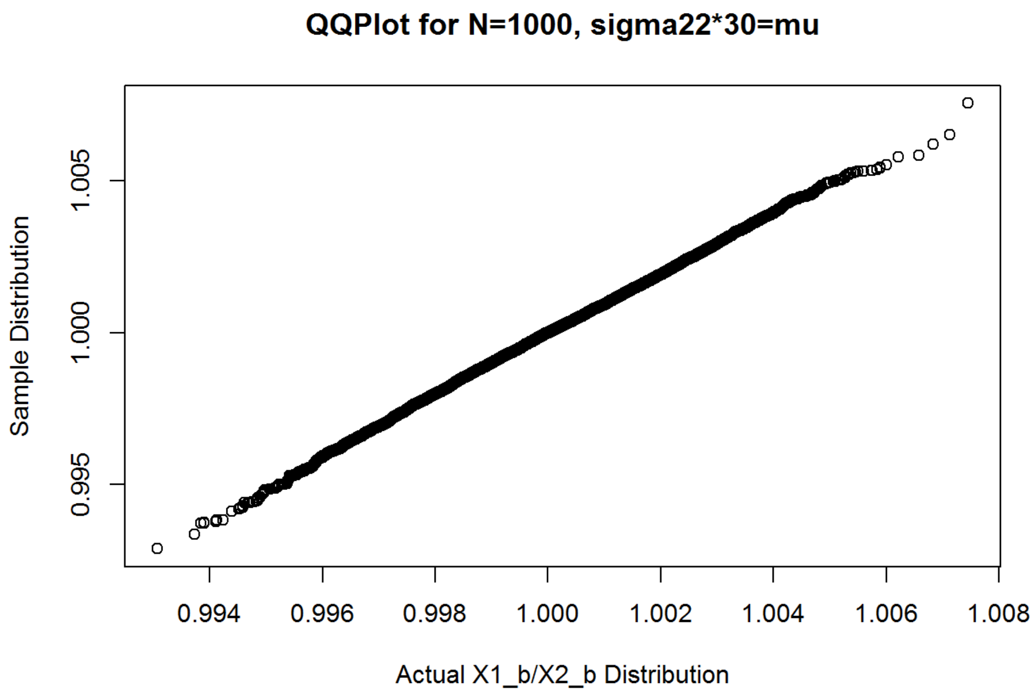
```
muVec = c(10, 10);
tranVec = matrix(c(1 / muVec[2], - muVec[1] / (muVec[2] ^ 2)), 2, 1);
sigmaAsympt = sqrt(t(tranVec) %*% covMat %*% tranVec);

seqN = 10;
asymptDraws = rnorm(numDraws) * sigmaAsympt / sqrt(seqN) + muVec[1] / muVec[2];
actDraws = getBivTransDraws(function(x, y) x / y, numDraws, seqN, muVec, covMat);
qqplot(actDraws,
asymptDraws,
main = "QQPlot for N=10, sigma22*30=mu",
xlab = "Actual X1_b/X2_b Distribution",
ylab = "Sample Distribution");
```




```
seqN = 100;
asymptDraws = rnorm(numDraws) * sigmaAsympt / sqrt(seqN) + muVec[1] / muVec[2];
actDraws = getBivTransDraws(function(x, y) x / y, numDraws, seqN, muVec, covMat);
qqplot(actDraws,
asymptDraws,
main = "QQPlot for N=100, sigma22*30=mu",
xlab = "Actual X1_b/X2_b Distribution",
ylab = "Sample Distribution");
```



```
seqN = 1000;
asymptDraws = rnorm(numDraws) * sigmaAsympt / sqrt(seqN) + muVec[1] / muVec[2];
actDraws = getBivTransDraws(function(x, y) x / y, numDraws, seqN, muVec, covMat);
qqplot(actDraws,
asymptDraws,
main = "QQPlot for N=1000, sigma22*30=mu",
xlab = "Actual X1_b/X2_b Distribution",
ylab = "Sample Distribution");
```



We conclude by noting that convergence is much faster in the second case.