Assignment-

Abheek Ghosh
140123047

August 28, 2017

1 Question a

Code for R

```
rm(list = ls())
d = read.table(”d-csp0108.txt”, header=TRUE)

# Calculating log returns
for (k in 2:3) {
   d[,k] = log(1 + d[,k]);
   # print(mean(d[,k]));
}
```

2 Question i

Code for R

```
rm(list = ls())
d = read.table(”d-csp0108.txt”, header=TRUE)
names = c(’C’, ’SP’)
# Calculating log returns
for (k in 2:3) {
    d[,k] = log(1 + d[,k]);
    cat(names[k-1], ”data, Mean =”, mean(d[,k]), ”\\\\\n”);
}

cat(”Yes, mean different from 0.\\\\\n”)

alpha = 0.05;
n = length(d[,1]);

for (k in 2:3) {
    mu = mean(d[,k]);
    sig = sd(d[,k]);

    # X = (d[,k] - mu)/sig;
    X = d[,k];
    # covProb = sum((qnorm(alpha/2) <= X) & (X <= qnorm(1 - alpha/2)));
    covProb = sum( ((X + sig*qnorm(alpha/2))/(1+sig) <= mu) & (mu <= (X + sig*qnorm(1 - alpha/2))/(1+sig)) );
    cat(names[k-1], ”data, Coverage Probability =”, covProb/n, ”\\\\\n”)
}
```

C data, Mean = -0.0008447986

SP data, Mean = -0.0001887621

Yes, mean different from 0.

C data, Coverage Probability = 0.9597215

SP data, Coverage Probability = 0.9507708

3 Question b

Code for R

```
rm(list = ls())
p = 0.3;
alpha = 0.05;

n = 1000;
M = c(20, 50, 100);

for (m in M) {
    covProb = 0;
    crossL = 0;
    crossU = 0;

    X = rbinom(m*n, size = 1, prob = p);
    p_mle = sum(X)/(n*m);
    # print(p_mle); print(length(X))

    for (i in 1:n) {
        Y = X[((i-1)*m + 1) : (i*m)];

        p_mean = mean(Y);
        # print((i-1)*m); print(p_mean)
        err = -qnorm(alpha/2)*sqrt(p_mean*(1-p_mean)/m);
        cil = p_mean - err;
        ciu = p_mean + err;

        if (cil < 0) {
            cil = 0;
            crossL = crossL + 1;
        }

        if (ciu > 1) {
            ciu = 1;
            crossU = crossU + 1;
        }

        if ((cil <= p_mle) && (p_mle <= ciu)) {
            covProb = covProb + 1;
        }
    }
    err = -qnorm(alpha/2)*sqrt(p_mle*(1-p_mle)/m);

    cat(sprintf(”\nBernoulli sample count = %d\np_mle = %f\n”, m, p_mle));
    cat(sprintf(”Confidence Interval = [%f, %f]\n”, p_mle-err, p_mle+err));
    cat(sprintf(”Coverage Probability = %f\n”, covProb/n));
    cat(sprintf("Number of instances of interval LOWER bound going outside parameter space = %d\n", crossL));
    cat(sprintf("Number of instances of interval UPPER bound going outside parameter space = %d\n", crossU));
}
```

Truncating the confidence interval from 0 is only partially justified. If we use central limit theorem and consider many IID samples then it approximates to normal distribution, but it doesn't fully become so for less number of samples of distributions other than normal. Distribution is bernoulli. It is evident from our result also: for 20 samples sometimes the interval goes beyond feasible region. As we are already doing approximation by using CLT for small samples, we can justify truncating the interval, it will not decrease the probability (confidence) of the interval. Other modifications can be by changing the coverage probability.

```
Bernoulli sample count = 20
p mle = 0.300900
Confidence Interval = [0.099892, 0.501908]
Coverage Probability = 0.938000
Number of instances of interval LOWER bound going outside parameter space = 112
Number of instances of interval UPPER bound going outside parameter space = 0


Bernoulli sample count = 50
p mle = 0.297940
Confidence Interval = [0.171171, 0.424709]
Coverage Probability = 0.943000
Number of instances of interval LOWER bound going outside parameter space = 0
Number of instances of interval UPPER bound going outside parameter space = 0


Bernoulli sample count = 100
p mle = 0.299270
Confidence Interval = [0.209516, 0.389024]
Coverage Probability = 0.966000
Number of instances of interval LOWER bound going outside parameter space = 0
Number of instances of interval UPPER bound going outside parameter space = 0
```