Problem 5

(a)

```
temp_val <- data.frame()
sn_save <- data.frame()
for(j in 1:10){
  for(i in 1:10000){
    temp_val[i,1] = sum(rnorm((10*j), mean = 0, sd = 1))
    temp_val[i,2] = abs(temp_val[i,1])
    }
  sn_save[j,1] <- (mean(temp_val[,2]))
}
row.names(sn_save) <- c("n=10","n=20","n=30","n=40",
                        "n=50","n=60","n=70","n=80","n=90","n=100")
colnames(sn_save) <- c("Expected Value of |Sn|")
sn_save
```

```
##       Expected Value of |Sn|
## n=10                2.563111
## n=20                3.564391
## n=30                4.363401
## n=40                5.030196
## n=50                5.570841
## n=60                6.256573
## n=70                6.699018
## n=80                7.101563
## n=90                7.523201
## n=100               7.941398
```

(b)

The distribution of S_n is based upon the value of n, where n takes the values $10, 20, 30 \cdots$

However, since we know that S_n is comprised of i.i.d. $X_i s$ which themselves are normally distributed, we can say that S_n is also normally distributed, where $\mu_{S_n} = 0$, and so,

$$S_n \sim \mathbb{N}(0, n)$$

Now, the 95% confidence interval for S_n is given by,

```
for(i in 1:10){
    print(paste("For n =", i*10,",the 95% is (", round(-1.96*sqrt(i*10),
                digits =4),",",round(1.96*sqrt(i*10),digits = 4),")"))
}
```

```
## [1] "For n = 10 ,the 95% is ( -6.1981 , 6.1981 )"
## [1] "For n = 20 ,the 95% is ( -8.7654 , 8.7654 )"
## [1] "For n = 30 ,the 95% is ( -10.7354 , 10.7354 )"
## [1] "For n = 40 ,the 95% is ( -12.3961 , 12.3961 )"
## [1] "For n = 50 ,the 95% is ( -13.8593 , 13.8593 )"
## [1] "For n = 60 ,the 95% is ( -15.1821 , 15.1821 )"
```

```
## [1] "For n = 70 ,the 95% is ( -16.3985 , 16.3985 )"
## [1] "For n = 80 ,the 95% is ( -17.5308 , 17.5308 )"
## [1] "For n = 90 ,the 95% is ( -18.5942 , 18.5942 )"
## [1] "For n = 100 ,the 95% is ( -19.6 , 19.6 )"
```

For the absolute value of S_n, the lower bound for the confidence interval will be 0, while the upper bound remains the same, and this will contain 95% of the values. This is because due to the 'folding' or the use of the absolute values, the 95% CI folds in, and the 95% upper bound remains the same while the lower bound becomes 0. So, the confidence interval for $|S_n|$ is given by,

```
for(i in 1:10){
    print(paste("For n =", i*10,",the 95% is ( 0,",round(1.96*sqrt(i*10),digits = 4),")"))
}
```

```
## [1] "For n = 10 ,the 95% is ( 0, 6.1981 )"
## [1] "For n = 20 ,the 95% is ( 0, 8.7654 )"
## [1] "For n = 30 ,the 95% is ( 0, 10.7354 )"
## [1] "For n = 40 ,the 95% is ( 0, 12.3961 )"
## [1] "For n = 50 ,the 95% is ( 0, 13.8593 )"
## [1] "For n = 60 ,the 95% is ( 0, 15.1821 )"
## [1] "For n = 70 ,the 95% is ( 0, 16.3985 )"
## [1] "For n = 80 ,the 95% is ( 0, 17.5308 )"
## [1] "For n = 90 ,the 95% is ( 0, 18.5942 )"
## [1] "For n = 100 ,the 95% is ( 0, 19.6 )"
```

(c)

We can verify our findings by editing our previous simulation to also record the CI bound values for each n.

```
temp_val <- data.frame()
sn_save <- data.frame()
for(j in 1:10){
  for(i in 1:10000){
    temp_val[i,1] = sum(rnorm((10*j), mean = 0, sd = 1))
    temp_val[i,2] = abs(temp_val[i,1])
  }
  temp_val[,2] <- temp_val[order(temp_val[,2]),2]
  sn_save[j,1] <- mean(temp_val[,2])
  sn_save[j,2] <- temp_val[1,2]
  sn_save[j,3] <- temp_val[9501,2]
}
row.names(sn_save) <- c("n=10","n=20","n=30","n=40",
                        "n=50","n=60","n=70","n=80","n=90","n=100")
colnames(sn_save) <- c("Expected Value of |Sn|","Lower Bound","Upper Bound")
sn_save
```

```
##       Expected Value of |Sn|  Lower Bound Upper Bound
## n=10                2.526063 1.694815e-04    6.198745
## n=20                3.596346 1.147185e-03    8.741096
## n=30                4.354548 2.368436e-03   10.740161
## n=40                5.031778 9.768985e-05   12.486107
## n=50                5.662284 9.894232e-04   14.023396
```

```
## n=60                 6.146714 5.049906e-04   15.100767
## n=70                 6.729930 4.747185e-04   16.520609
## n=80                 6.993729 3.283182e-04   17.147619
## n=90                 7.751934 1.707308e-03   18.783362
## n=100                7.937878 4.512142e-04   19.262166
```

We find that our bounds found through the simulation are very similar to the theoretical bounds.