Assignment_3

1a. Calculate the mean absolute deviation of (3.1, 2.8, 4.0, 3.5, 3.3).

```
D <- c(3.1, 2.8, 4.0, 3.5, 3.3)
print((1/length(D))*sum(abs(D - mean(D))))
```

```
## [1] 0.328
```

1b. Develop a function to calculate the mean absolute deviation of a vector. The input is a numeric vector, and the output is a scalar.

```
MAD <- function(D)
{
  n <- length(D)
  D <- abs(D - mean(D))
  return((1/n)*sum(D))
}
```

1c. Generate 100,000 independent standard normal random numbers by the following code. Then, calculate the mean absolute deviation of Y by the function developed in (b).

```
set.seed(123)
Y <- rnorm(100000)
MAD(Y)
```

```
## [1] 0.7980813
```

2. Daily returns of a stock X in percentage for 50 days are available in Canvas (Return.csv). We are analyzing the risk to invest in the stock X. Its sample standard deviation $\hat{\sigma}$ is 1.5598, but you want to see how reliable $\hat{\sigma}$ is. Use bootstrap method to estimate the standard devi- ation of $\hat{\sigma}$.

```
R <- read.csv("Return.csv",header = T)
M <- numeric(100)

for(i in 1:length(M))
{
  M[i] <- sd(sample(R$x,replace=T))
}

sd(M)
```

```
## [1] 0.3784202
```

3. Generate 10,000 t-random variables with degrees of freedom 4 in two ways: by using the ”rt” function, and by using the ”rnorm” function, and compare the means and standard deviations.

```r
TCOMPAREMEAN <- numeric(100)
TCOMPARESD <- numeric(100)
NCOMPAREMEAN <- numeric(100)
NCOMPARESD <- numeric(100)

for(i in 1:100)
{
  TRANDOM <- rt(10000,4)
  TCOMPAREMEAN[i] <- mean(TRANDOM)
  TCOMPARESD[i] <- sd(TRANDOM)

  NORMAL <- (rnorm(10000))/(sqrt((sum((rnorm(4))^2))/4))
  NCOMPAREMEAN[i] <- mean(NORMAL)
  NCOMPARESD[i] <- sd(NORMAL)
}

print("Average mean and standard deviation of rt")
```

```
## [1] "Average mean and standard deviation of rt"
```

```r
mean(TCOMPAREMEAN)
```

```
## [1] 4.590314e-05
```

```r
mean(TCOMPARESD)
```

```
## [1] 1.411912
```

```r
print("Average mean and standard deviation of rnorm")
```

```
## [1] "Average mean and standard deviation of rnorm"
```

```r
mean(NCOMPAREMEAN)
```

```
## [1] 0.00010487
```

```r
mean(NCOMPARESD)
```

```
## [1] 1.224348
```

```r
#It is clear that rnorm is more accurate since standard deviation is lower on
average.
```