UKSta18_Liang_Ex03

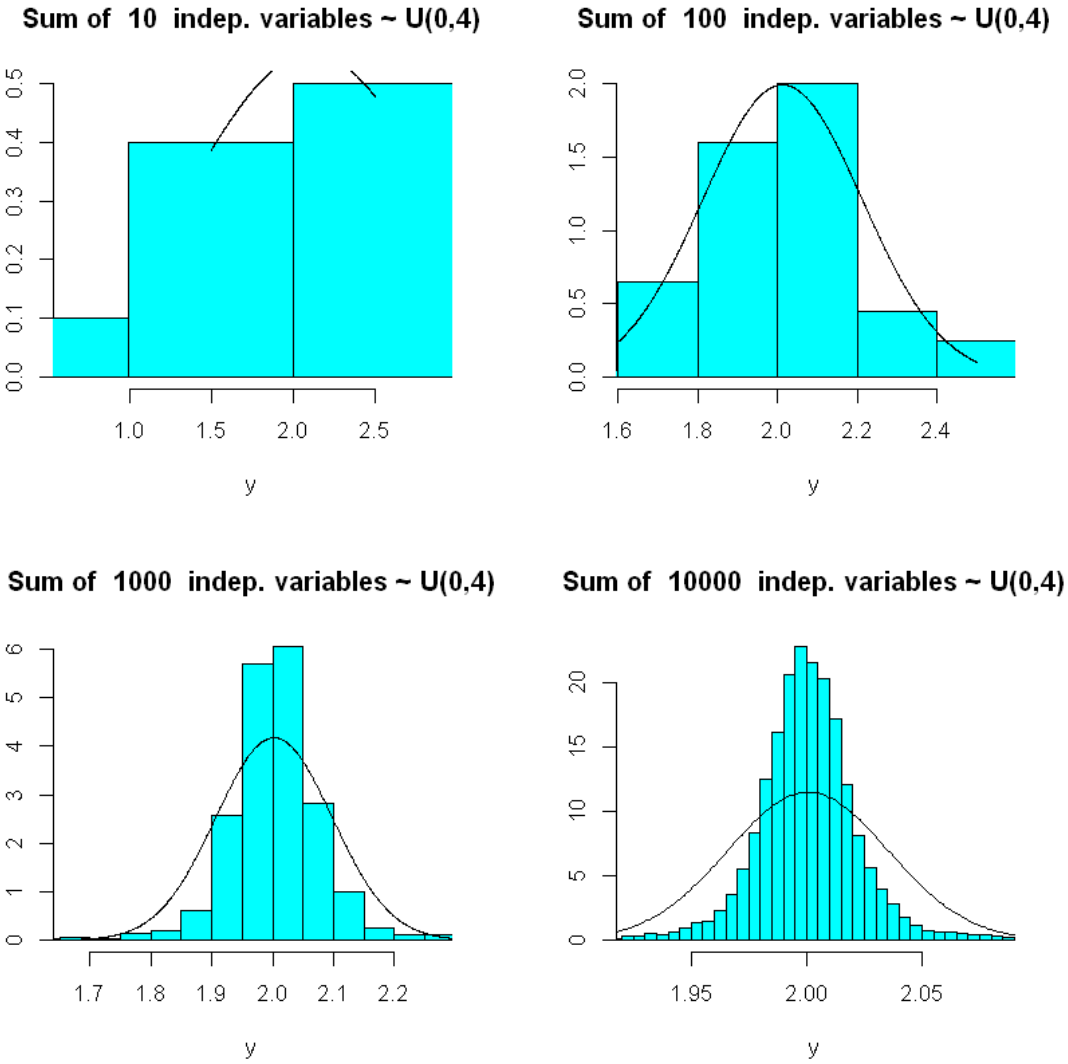
August 9, 2018

1 A graphical demonstration of the central limit theorem

Average a number of N of very non-Gaussian uniform distributions U(0,4).

- a: We plot a series of histograms demonstrating the convergence to Gaussian distribution with increasing number N.

```
In [24]: # Instead of hist we use truehist() from MASS to get the right normalization
         library(MASS)
         par(mfrow=c(2,2))
         for (n in c(10, 100, 1000, 10000)){  # for N = 10, 100, 1000, 10000
             xi <- matrix(NA, n, n)
             for (i in 1:n){
                 # row_i has i independent variables ~ U(0,4)
                 xi[i,] <- c(runif(i,0,4),rep(NA, n-i))
             }
             # y on i-th row is the sum of i independent variables
             y <- rowMeans(xi,na.rm = TRUE)
             # in order to get reasonable axis-scaling we choose quantiles at 1% and 99% to
             # plot datas between them
             q <- quantile(y, probs = c(0.01, 0.99))
             names(q) <- NULL
             truehist(y, main=paste("Sum of ",n, " indep. variables ~ U(0,4)"), xlim=q)
             # Overlay the expected normal distribution
             lines(seq(1.5, 2.5, 0.001), dnorm(seq(1.5, 2.5, 0.001),mean(y), sd(y)), col="blac
         }
```



- b: How large must be N to get a reasonable correspondence? How could one quantify the convergence?

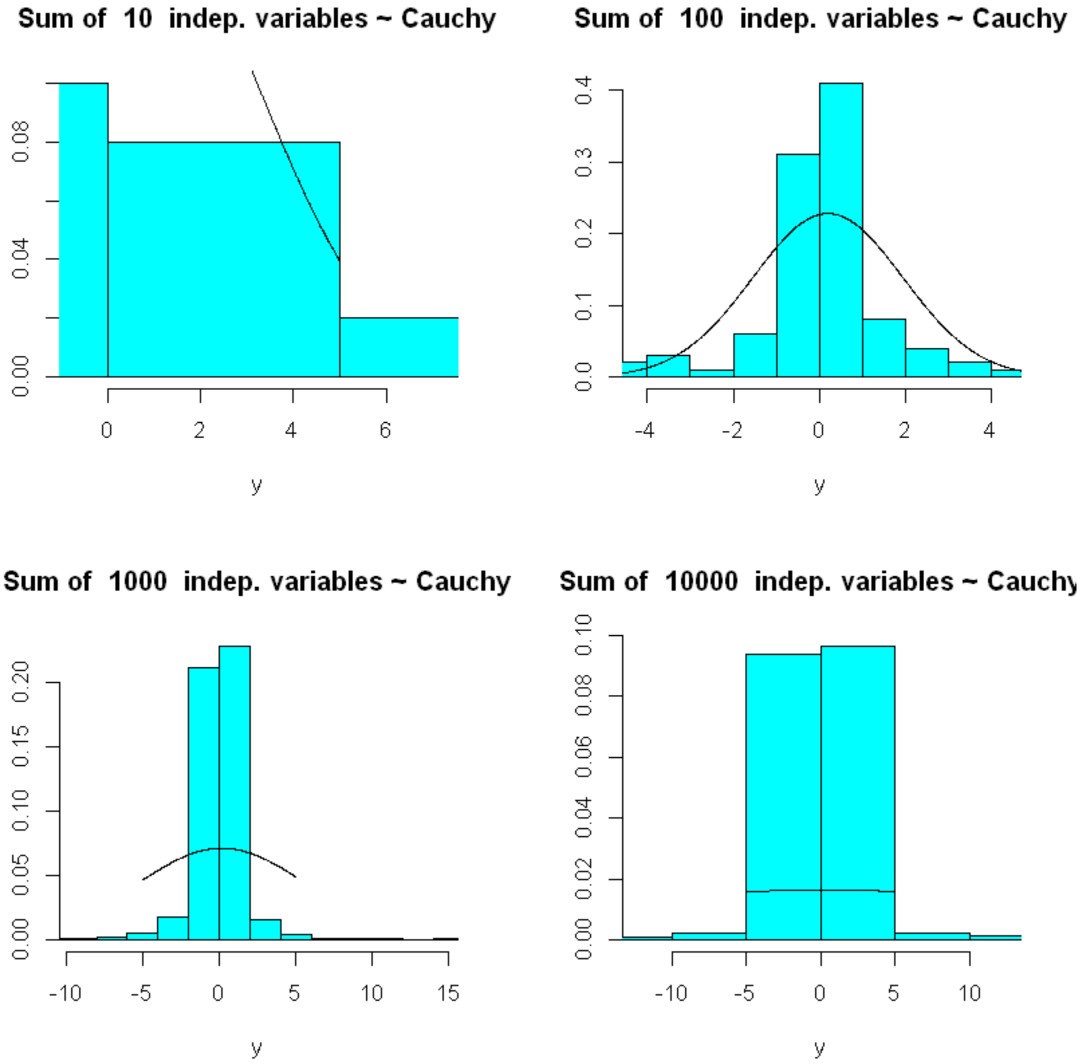
  One can see that the form of histograms resembling a normal distribution when N is around 10000. In that sense we already have a reasonable correspondance. The reason that the curve is not good fitted to the histogram might be that there's still some problem with normalization.

  To quantify the convergence one could calculate the square sum of the difference between the frequency corresponding to the middle value of each bar and the corresponding value on the fitted Gaussian curve. If it's decreasing with increasing N then the convergence is shown.

- c: Now we do the same using a Cauchy distribution.

In [44]:

```
par(mfrow=c(2,2))
for (n in c(10, 100, 1000, 10000)){
    xi <- matrix(NA, n, n)
    for (i in 1:n){
        xi[i,] <- c(rcauchy(i, 0, 0.4),rep(NA, n-i))
    }
    y <- rowMeans(xi,na.rm = TRUE)
    q <- quantile(y, probs = c(0.01, 0.99))
    names(q) <- NULL
    truehist(y, main=paste("Sum of ",n, " indep. variables ~ Cauchy"), xlim=q)
    lines(seq(-5, 5, 0.01), dnorm(seq(-5, 5, 0.01),mean(y), sd(y)), col="black")
}
```



An explanation for the behavior: ...

2 Testing mean, median, and trimean for efficiency and robustness

Commonly, mean and median are used as estimators for the mean of a symmetric distribution. Another, less known one, is the trimean which is a weighted average of the distribution's median and first and third quartiles

$$\hat{\mu}_{TM} \equiv \frac{1}{4}(Q_1 + 2Q_2 + Q_3)$$

We test the 3 estimators for a normal distribution N(0,1) with a sample size of 20.

- a: First we construct an R-function which calculates the trimean.

```
In [39]: trimean <- function(x){
             # x is a matrix
             q <- apply(x, 1, quantile)
             names(q) <- NULL    # avoid getting "25%" in result
             return((q[2,]+2*q[3,]+q[4,])/4)
         }
         # test
         print(trimean(matrix(1:9,3,3)))
```

```
[1] 4 5 6
```

- b: Then we run Monte Carlo tests to check that the estimators are indeed all unbiased.

```
In [40]: # normal distribution, mean = 0
         a <- matrix(rnorm(20*10), 20, 10)
         cat("Bias of mean estimator, 10 trials\n")
         cat("==================================\n")
         cat("Bias with mean: ", mean(rowMeans(a)), "\n")
         Median <- apply(a, 1, median)
         cat("Bias with median: ", mean(Median), "\n")
         cat("Bias with trimean: ", mean(trimean(a)), "\n")
         cat("-----------------------------------\n")
         a <- matrix(rnorm(20*10000), 20, 1000)
         cat("Bias of mean estimator, 1000 trials\n")
         cat("==================================\n")
         cat("Bias with mean: ", mean(rowMeans(a)), "\n")
         Median <- apply(a, 1, median)
         cat("Bias with median: ", mean(Median), "\n")
         cat("Bias with trimean: ", mean(trimean(a)), "\n")
         cat("-----------------------------------\n")
         a <- matrix(rnorm(20*100000), 20, 100000)
         cat("Bias of mean estimator, 100000 trials\n")
         cat("==================================\n")
         cat("Bias with mean: ", mean(rowMeans(a)), "\n")
         Median <- apply(a, 1, median)
         cat("Bias with median: ", mean(Median), "\n")
         cat("Bias with trimean: ", mean(trimean(a)), "\n")
```

```
Bias of mean estimator, 10 trials
==================================
Bias with mean:  -0.09608443
Bias with median:  -0.09344324
Bias with trimean:  -0.07792757
-----------------------------------
Bias of mean estimator, 1000 trials
==================================
Bias with mean:  -0.002563217
Bias with median:  -0.005331312
Bias with trimean:  -0.003005625
-----------------------------------
Bias of mean estimator, 100000 trials
==================================
Bias with mean:  -0.001153126
Bias with median:  -0.000955006
Bias with trimean:  -0.001202874
```

We see from the decreasing biases calculated by running 10, 1000 and 100000 trials that the MC results are statistically significant by varying the number of Monte Carlo trials. After running 100000 trials the bias is small enough so that it's fair to say that the 3 estimators are unbiased.

- c: Now we test the efficiency of the 3 estimators.

```
In [41]: cat("Efficiency of mean estimator, 100000 trials\n")
         cat("=========================\n")
         cat("Efficiency with mean: ", mean(rowMeans(a)^2), "\n")
         Median <- apply(a, 1, median)
         cat("Efficiency with median: ", mean(Median^2), "\n")
         cat("Efficiency with trimean: ", mean(trimean(a)^2), "\n")
```

```
Efficiency of mean estimator, 100000 trials
=========================
Efficiency with mean:  6.501178e-06
Efficiency with median:  1.467145e-05
Efficiency with trimean:  1.028209e-05
```

Clearly that the estimator mean performs best. The ratio of the efficiency of estimator median to the best one: $\frac{1.186}{1.039} \approx 1.14$, The ratio of the efficiency of estimator trimean to the best one: $\frac{1.220}{1.039} \approx 1.17$. It shows us that the three estimators have almost the same efficiency and the estimator mean is relatively better. * d: Now we add one (rather extreme) outlier drawn from a uniform distribution U(20,20) and test the robustness by looking at the mean-square error in the cases.

```
In [42]: b <- matrix(runif(20*100000,-20,20),20, 100000)
         # add outlier as the last element
         b[,1] <- 100000
         cat("Efficiency of mean estimator with an outlier, 100000 trials\n")
         cat("=========================\n")
         cat("Efficiency with mean: ", mean(rowMeans(b)^2), "\n")
         Median <- apply(b, 1, median)
         cat("Efficiency with median: ", mean(Median^2), "\n")
         cat("Efficiency with trimean: ", mean(trimean(b)^2), "\n")
```

```
Efficiency of mean estimator with an outlier, 100000 trials
=========================
Efficiency with mean:  1.005677
Efficiency with median:  0.005425619
Efficiency with trimean:  0.003112881
```

Clearly that the estimator trimean performs best. The ratio of the efficiency of estimator mean to the best one: $\frac{1.0054}{0.0015} \approx 670.27$, The ratio of the efficiency of estimator median to the best one: $\frac{0.0029}{0.0015} \approx 1.93$. It shows us that the estimator mean performs especially bad when the data has one (extreme) outlier.