Week2AssignmentPart3

Part 3. Invent a random number generator

Think about possible sources of true or pseudo-random sequences of {0,1} and choose one or two of them. Conduct the tests described in the previous section.

Part1 Description of your random number generator

My sequence is taken from a MLB pitcher (Jake Arrieta) career statistics for walks and strikeouts per game. For each game I tabulated a ratio of walks divided by strikeouts, and removed the NA values(for when he had zero strikeouts) as well as the values greater than 1 for when he had more walks than strikeouts. We are left with 128 datapoints between 0 and 1.

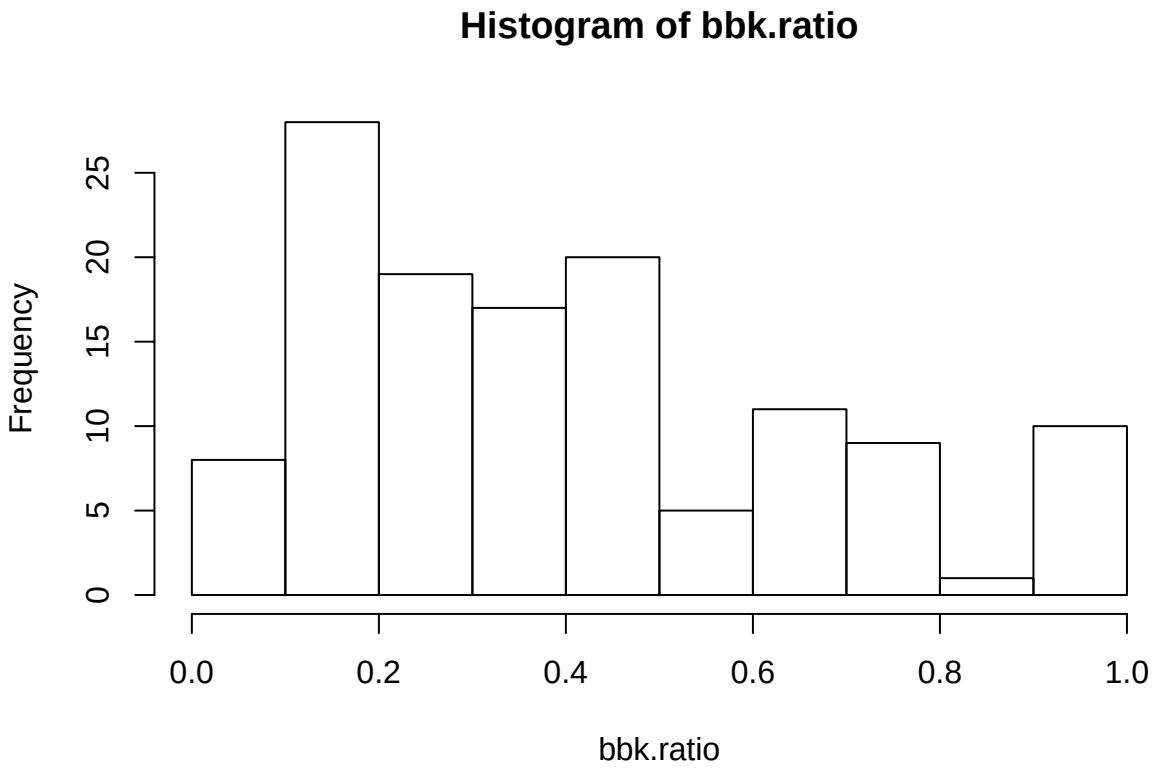
Part2 Generated sequence

```
setwd("C:/Users/JohntheGreat/Documents/MSCA/StatisticalAnalysis/Week2/Assignments")
ratio <- read.csv(file = "ArrietaRatio.csv", header = F)
bbk.ratio <- ratio$V3
print(bbk.ratio)
```

```
##   [1] 0.07692308 0.08333333 0.08333333 0.09090909 0.09090909 0.10000000
##   [7] 0.10000000 0.10000000 0.11111111 0.11111111 0.11111111 0.12500000
##  [13] 0.12500000 0.12500000 0.12500000 0.14285714 0.14285714 0.14285714
##  [19] 0.14285714 0.14285714 0.16666667 0.16666667 0.16666667 0.16666667
##  [25] 0.16666667 0.16666667 0.18181818 0.20000000 0.20000000 0.20000000
##  [31] 0.20000000 0.20000000 0.20000000 0.20000000 0.20000000 0.20000000
##  [37] 0.22222222 0.25000000 0.25000000 0.25000000 0.25000000 0.25000000
##  [43] 0.25000000 0.25000000 0.25000000 0.25000000 0.27272727 0.28571429
##  [49] 0.28571429 0.28571429 0.28571429 0.28571429 0.28571429 0.28571429
##  [55] 0.30000000 0.33333333 0.33333333 0.33333333 0.33333333 0.33333333
##  [61] 0.33333333 0.33333333 0.33333333 0.37500000 0.37500000 0.38368247
##  [67] 0.40000000 0.40000000 0.40000000 0.40000000 0.40000000 0.40000000
##  [73] 0.42857143 0.42857143 0.42857143 0.50000000 0.50000000 0.50000000
##  [79] 0.50000000 0.50000000 0.50000000 0.50000000 0.50000000 0.50000000
##  [85] 0.50000000 0.50000000 0.50000000 0.50000000 0.50000000 0.50000000
##  [91] 0.50000000 0.50000000 0.57142857 0.60000000 0.60000000 0.60000000
##  [97] 0.60000000 0.66666667 0.66666667 0.66666667 0.66666667 0.66666667
## [103] 0.66666667 0.66666667 0.66666667 0.66666667 0.66666667 0.66666667
## [109] 0.71428571 0.75000000 0.75000000 0.75000000 0.75000000 0.75000000
## [115] 0.75000000 0.80000000 0.80000000 0.83333333 1.00000000 1.00000000
## [121] 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000 1.00000000
## [127] 1.00000000 1.00000000
```

Part3 Uniformity test

```
bbk.ratio.histogram<-hist(bbk.ratio)
```




```
bbk.ratio.histogram
```

```
## $breaks
##  [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
##
## $counts
##  [1]  8 28 19 17 20  5 11  9  1 10
##
## $density
##  [1] 0.625000 2.187500 1.484375 1.328125 1.562500 0.390625 0.859375
##  [8] 0.703125 0.078125 0.781250
##
## $mids
##  [1] 0.05 0.15 0.25 0.35 0.45 0.55 0.65 0.75 0.85 0.95
##
## $xname
## [1] "bbk.ratio"
##
## $equidist
## [1] TRUE
```

```
##
## attr(,"class")
## [1] "histogram"
```

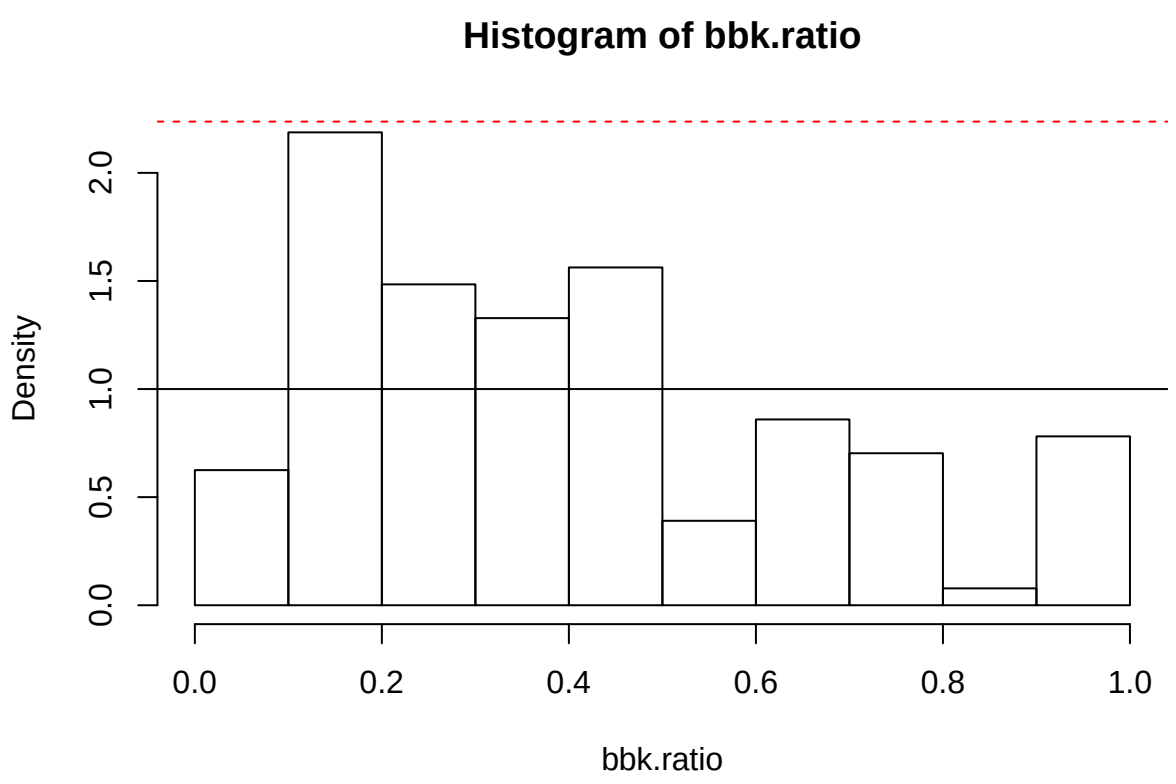
```
(bbk.ratio.histogram.mean<-mean(bbk.ratio.histogram$density))
```

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

```
(bbk.ratio.histogram.sd<-sd(bbk.ratio.histogram$density))
```

```
## [1] 0.631262
```

```
plot(bbk.ratio.histogram,freq=FALSE)
abline(h=bbk.ratio.histogram.mean)
abline(h=bbk.ratio.histogram.mean+1.96*bbk.ratio.histogram.sd,col="red",lty=2)
abline(h=bbk.ratio.histogram.mean-1.96*bbk.ratio.histogram.sd,col="red",lty=2)
```




```
(bbk.ratio.mean<-mean(bbk.ratio))
```

```
## [1] 0.4237362
```

```r
(bbk.ratio.variance<-var(bbk.ratio))
```

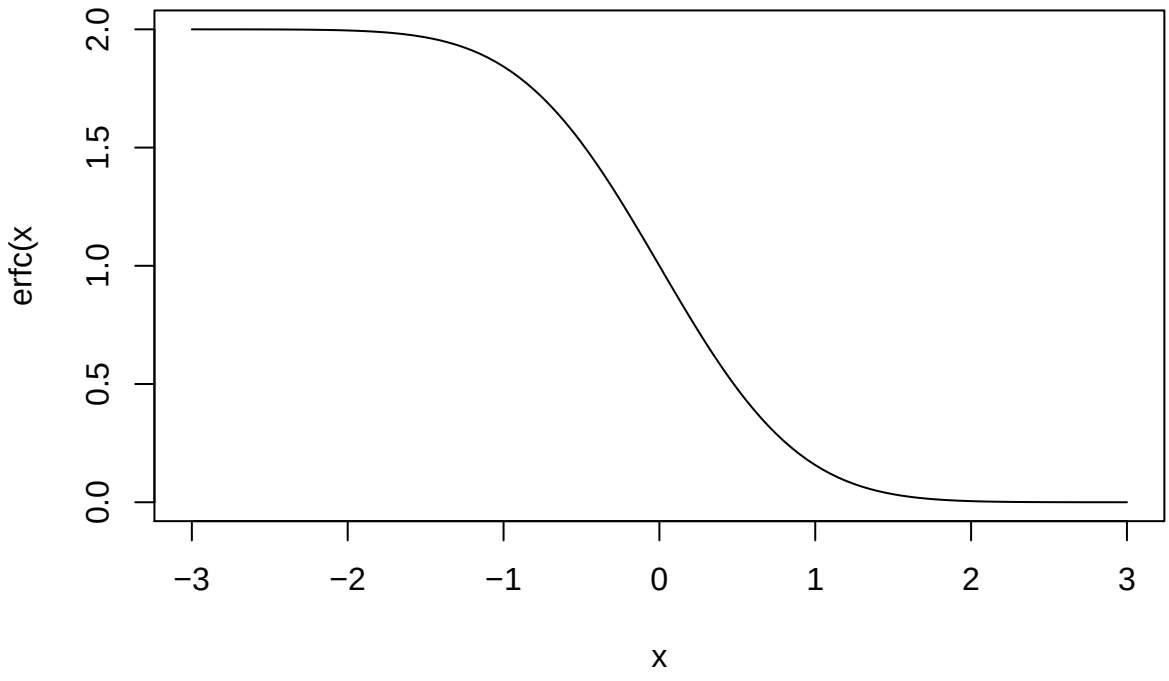
```
## [1] 0.06918158
```

```r
summary(bbk.ratio)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.07692 0.20000 0.37500 0.42370 0.60000 1.00000
```

Part4 frequency test

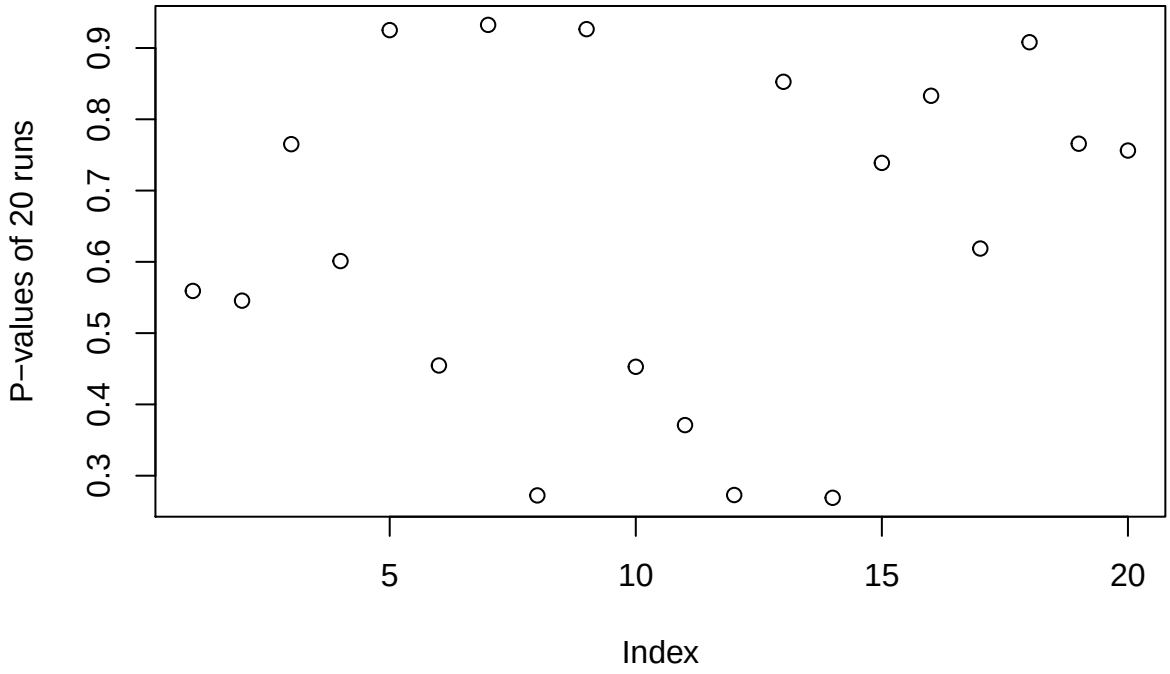
```r
##convert bbk.ratio to binary sample bbk.bin
##For now, use runif for set of {0,1}
nFlips <- 1000
set.seed(15)
bbk.bin<-runif(1000,0,1)

bbk.bin.plusminus1 <- (bbk.bin - .5)*2

erf <- function(x) 2 * pnorm(x * sqrt(2)) - 1
erfc <- function(x) 2 * pnorm(x * sqrt(2), lower = FALSE)

plot(seq(from=-3,to=3,by=.05),erfc(seq(from=-3,to=3,by=.05)),type="l",xlab="x",ylab="erfc(x")
```



```
erfc(abs(sum(bbk.bin.plusminus1)/sqrt(2*nFlips)))
```

```
## [1] 0.3060178
```

```
plot(erfc(abs(apply(matrix(bbk.bin.plusminus1,ncol=50),1,sum))/sqrt(2*50)),ylab="P-values of 20 runs")
```




```
sum(erfc(abs(apply(matrix(bbk.bin.plusminus1,ncol=50),1,sum))/sqrt(2*50))<=.01)
```

```
## [1] 0
```

Part5 Turning point test

```
library(randtests)
turning.point.test(bbk.ratio)
```

```
##
##  Turning Point Test
##
## data:  bbk.ratio
## statistic = -8.1875, n = 29, p-value = 2.668e-16
## alternative hypothesis: non randomness
```