hw2.R

mwilde

Fri Oct 13 12:59:10 2017

```
# HW2
# matt wilde

#########################################################
# problem 1
#########################################################

# a)
n = c(10, 100, 1000, 10000, 100000)
for (i in n){
  dist = rpois(i, lambda = 2)
  print(paste("n = ",i, "mean = ", mean(dist), "var = ", var(dist)))
}
```

```
## [1] "n =  10 mean =  1.6 var =  1.6"
## [1] "n =  100 mean =  1.94 var =  1.73737373737373"
## [1] "n =  1000 mean =  2.006 var =  2.01397797797798"
## [1] "n =  10000 mean =  2.0098 var =  2.00510447044704"
## [1] "n =  1e+05 mean =  2.00248 var =  2.00539390353903"
```

```
# b)
n = 200
p = 0.2
X = rbinom(1, size = 200, prob = 0.2)
print(paste("E(X) = np =", n*p, mean(X)))
```

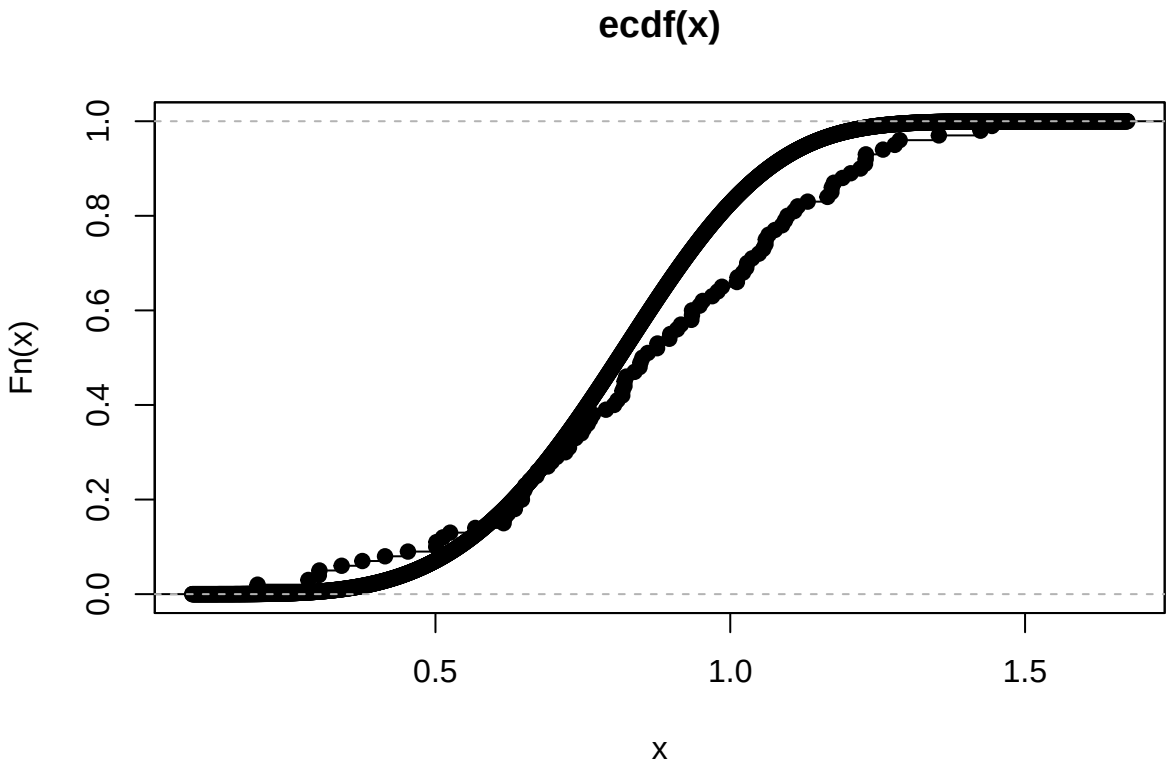
```
## [1] "E(X) = np = 40 47"
```

```
for (i in 1:5){
  dist = rbinom(10^i, size = 200, prob = 0.2)
  print(paste("n = ",10^i, "mean = ", mean(dist), "var = ", var(dist)))
}
```

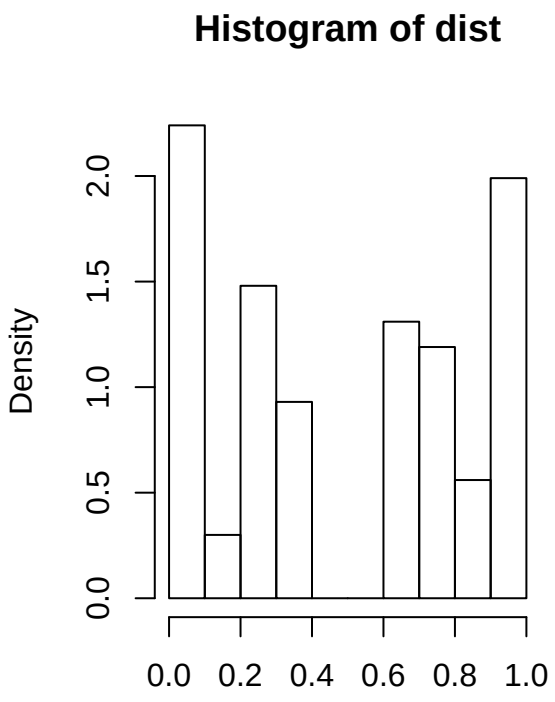
```
## [1] "n =  10 mean =  41 var =  34.2222222222222"
## [1] "n =  100 mean =  40.71 var =  35.8645454545455"
## [1] "n =  1000 mean =  40.375 var =  32.4908658658659"
## [1] "n =  10000 mean =  40.0146 var =  32.0479916391639"
## [1] "n =  1e+05 mean =  40.00115 var =  32.002688704387"
```

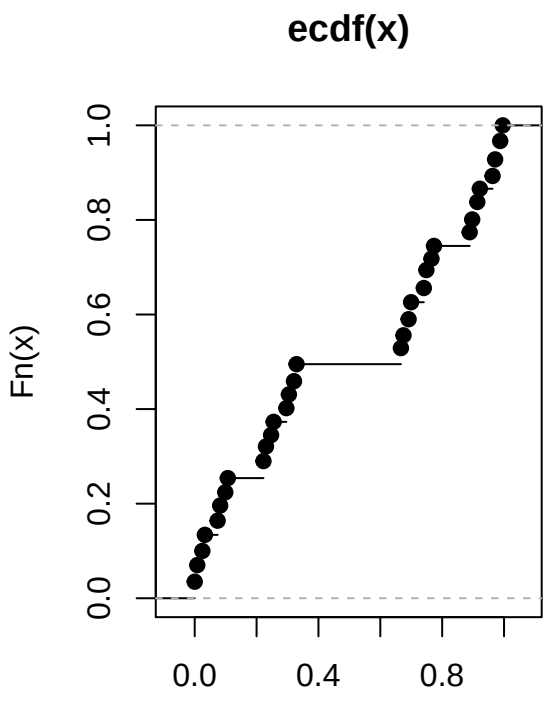
```
#########################################################
# problem 2
#########################################################
# a)
X = rweibull(100, shape = 4, scale = 1)


# b)
# Use the pweibull or qweibull functions to add the CDF for a
# Weibull distributed variable with shape parameter 4 (and scale 1) to your plot.
```

```r
P = pweibull(seq(0,2,0.001), shape = 4)

plot(P, xlab = "", ylab = "", xaxt = "n", yaxt = "n")
par(new = TRUE)
plot.ecdf(X)
```



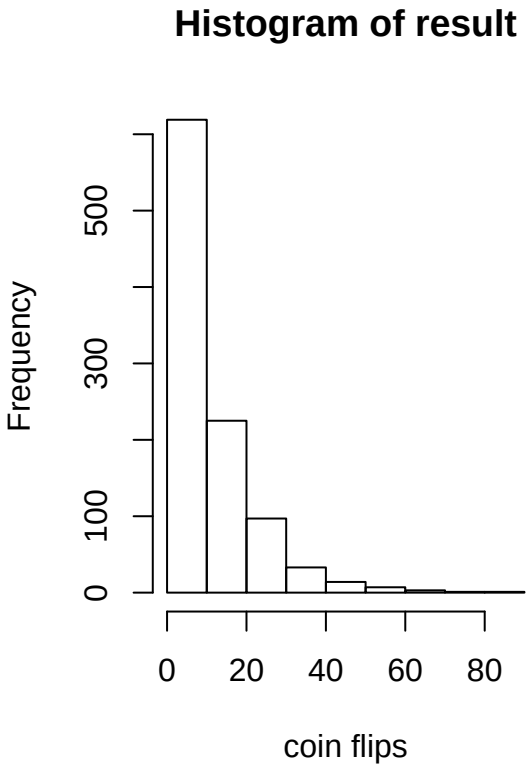

```r
#########################################################
# problem 3
#########################################################
# c)


# plots.
# how many rolls of the dice?
k = c(1:5)
p = c(2/3^k)

# what does this look like after 10000 trials of computing X5
dist = c()
for (i in 1:1000){
  # points after 5 flips
  X5 = sum(rbinom(5, 1, 0.5)*p)
  dist[i] = X5
}
par(mfrow = c(1,2))
hist(dist, probability = TRUE, xlab = 'X5')
plot.ecdf(dist)
```






```
#########################################################
# problem 5
#########################################################
result = c()
for (i in 0:1000){
  success = FALSE
  k = 0
  while (!success){
    # flip two coins
    p = rbinom(1,1,0.1) + rbinom(1,1,0.1)
    k = k + 1

    # if 2 coins result in a combo of H and T
    if (p == 1){
      # print(k)
      # assign nummber of tries to get HT/TH to result vector
      result[i] = 2 * k

      # increment result index
      i = i + 1

      #exit while loop
      success = TRUE
    }
  }
}

hist(result, xlab = 'coin flips')
print(mean(result))
```

```
## [1] 11.634
```


Histogram of result
Frequency
0
100
300
500
0
20
40
60
80
coin flips