

STA 220 - Data and Web Technologies for Data Analysis - Lab 3

Consider the following set of probabilities 0.5, 0.6, 0.7, 0.8 and 0.9 for the example in Lecture 3.

- What should we expect to observe?
- What happen if we consider 100 heads in a row?

Hint

Use the functions of Lecture 3:

```
simtosses <- function(nsim, ntoss, probH = 1/2) {  
  matrix(sample(0:1, ntoss * nsim, replace = TRUE,  
              prob = c(1 - probH, probH)),  
         ncol = nsim)  
}  
# ntoss by nsim matrix
```

```
fmaxrl <- function(x, outcome = NULL) {  
  xr <- rle(x) # run length encoding  
  if (!is.null(outcome)) {# if outcome is not NULL  
    is_outcome <- (xr$values == outcome)  
    xr$lengths <- xr$lengths[is_outcome]  
  }  
  if (length(xr$lengths) == 0L) 0 else max(xr$lengths)  
  # if length is not 0, return the maximum number of consecutive heads or tails  
}
```

```
nsim <- 10000  
ntoss <- 100  
pH <- c(0.5, 0.6, 0.7, 0.8, 0.9) # probabilities of head  
res <- matrix(nrow = nsim, ncol = length(pH))  
colnames(res) <- pH  
for (i in seq_along(pH)) {  
  res[, i] <- apply(simtosses(nsim, ntoss, probH = pH[i]), 2, fmaxrl)  
}  
muhat <- apply(res, 2, mean) # what should we expect to observe  
sigmahat <- apply(res, 2, sd) # uncertainty  
se.muhat <- sigmahat / sqrt(nsim)  
round(rbind(muhat, se.muhat), 3) # rounds the values in its first argument to the specified number of de  
##           0.5    0.6    0.7    0.8    0.9  
## muhat    6.984  7.981 10.686 15.625 26.802  
## se.muhat 0.018 0.023 0.034 0.052 0.095
```

We can use the function apply

```
apply(res, 2, quantile, probs = c(0, 0.05, 0.25, 0.5, 0.75, 0.95, 1))  
##           0.5 0.6 0.7 0.8 0.9
```

```
## 0%      3  4  4  6  8
## 5%      5  5  6  9 15
## 25%     6  6  8 12 20
## 50%     7  8 10 15 25
## 75%     8  9 12 18 32
## 95%    10 12 17 26 45
## 100%   19 25 33 49 85
# more succinct and nice output
# if using loop, we have
result <- matrix(nrow = 7, ncol = 5)
for(i in 1:ncol(res)){
  result[, i] <- quantile(res[, i], probs = c(0, 0.05, 0.25, 0.5, 0.75, 0.95, 1))
}
result
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    3    4    4    6    8
## [2,]    5    5    6    9   15
## [3,]    6    6    8   12   20
## [4,]    7    8   10   15   25
## [5,]    8    9   12   18   32
## [6,]   10   12   17   26   45
## [7,]   19   25   33   49   85

opar <- par(mar = c(5.1, 4.1, 1.1, 1.1))# set graph margins using par()
boxplot(res, horizontal = TRUE,
        xlab = "Maximum Run Length", ylab = "Probability of Heads")
par(opar)# reset the graph margins back to the default values
# type par() in console to print all the default graphical parameters
```

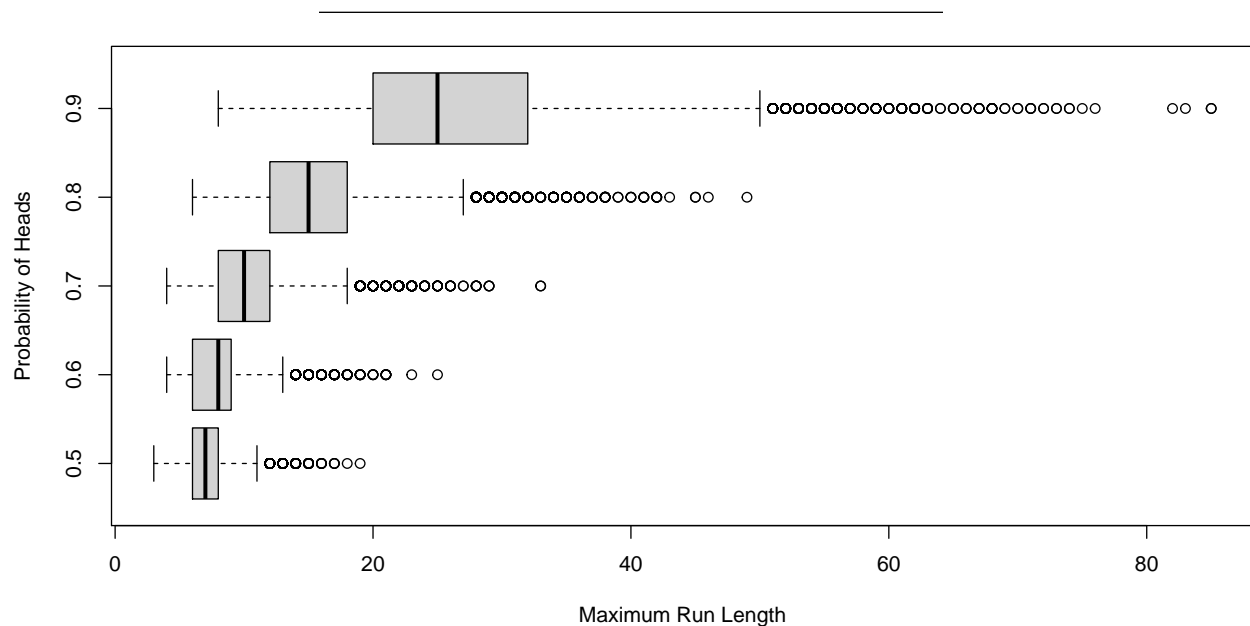


Figure 1: Maximum run lengths in 10,000 simulations of 100 tosses of a biased coin.

If we consider 100 heads in a row

With the probability of heads set to 0.9, the probability of getting 100 heads in 100 (independent) tosses of the (biased) coin is pretty small.

```
0.9^100  
## [1] 2.65614e-05
```

However, if we repeat the experiment 10,000 times, the probability that we get 100 heads at least once is not so small.

```
1 - (1 - 0.9^100)^10000  
## [1] 0.2332677
```