

```

pvec = seq(0, 1, by = 1 / 3)
qvec = seq(0, 1, by = 1 / 3)
pn = length(pvec)

n = 100
Y = rep(0, n)
res = matrix(data = NA, ncol = 10, nrow = n) # Длины pvec qvec
одинаковы и равны n
columnIndex = 1
for (i in (1:pn)) {
  for (j in (1:pn)) {
    p = pvec[i]
    q = qvec[j]
    if (p + q <= 1) {
      X = sample(
        c("камень", "Ножницы", "Бумага"),
        size = n,
        replace = TRUE,
        prob = c(p, q, 1 - p - q)
      )
      # eta1 = runif(n)
      # eta2 = runif(n)
      eta1 = rbeta(n, 1, 2)
      eta2 = rbeta(n, 1, 2)
      for (k in (1:n)) {
        first = min(eta1[k], eta2[k])
        second = max(eta1[k], eta2[k])
        Y[k] = sample(
          c("камень", "Ножницы", "Бумага"),
          size = 1,
          replace = TRUE,
          prob = c(first, second - first, 1.0 - second)
        )
        result = 0
        if ((X[k] == "камень" &&
          Y[k] == "Ножницы") ||
          (X[k] == "Ножницы" &&
          Y[k] == "Бумага") || (X[k] == "Бумага" && Y[k] ==
"камень")) {
          result = 1
        }
        res[k, columnIndex] = result
      }
      print(columnIndex)
      columnIndex = columnIndex + 1
    }
  }
}

for (i in (1:10)) {
  print(sum(res[, i]) / n * 100)
}

```

```
# При равномерном распределении вероятности выигрыша в процентах
# 1] 31
# [1] 30
# [1] 37
# [1] 41
# [1] 31
# [1] 36
# [1] 36
# [1] 36
# [1] 24
# [1] 32
#
# При бета распределении вероятность выигрыша в процентах
#
# [1] 27
# [1] 37
# [1] 38
# [1] 58 (p = 0 q = 1) - ма
# [1] 19
# [1] 36
# [1] 45
# [1] 17
# [1] 33
# [1] 31
```