Homework 3 - Problem B

Rajat Mahesh Gupta

2024-12-04

Here you will develop "d,p,q,r" functions for a certain distribution family, in the sense of e.g. Sec. 4.4.1.

- We'll call the family "accum" for "accumulate." The setting is that of repeatedly rolling a pair of dice. The random variable X is the number of rolls needed to achieve an accumulated total of at least k dots. So for instance the support of X ranges from ceiling(k/12) to ceiling(k/2). This is a one-parameter family.
- The call forms will be

```
daccum(i,k)
paccum(i,k)
qaccum(m,k)
raccum(nreps,k)
```

- The 'd', 'p' and 'q' functions must be exact, i.e. not computed via simulation, but a recursive call is fine. Finding 'p' and 'q' from 'd' is fine.
- For 'q', note the comment following (4.31).

```r
# Probability mass function for the accum distribution
daccum <- function(i, k) {
  if (i < ceiling(k / 12) || i > ceiling(k / 2)) {
    return(0)
  }
  if (i == ceiling(k / 12)) {
    return(1)
  }
  return(daccum(i - 1, k) * (1 - paccum(i - 1, k)))
}

# Cumulative distribution function for the accum distribution
paccum <- function(i, k) {
  if (i < ceiling(k / 12)) {
    return(0)
  }
  if (i >= ceiling(k / 2)) {
    return(1)
  }
  return(sum(sapply(ceiling(k / 12):i, function(j) daccum(j, k))))
}

# Quantile function for the accum distribution
```

```r
qaccum <- function(m, k) {
  p <- m / 100
  for (i in ceiling(k / 12):ceiling(k / 2)) {
    if (paccum(i, k) >= p) {
      return(i)
    }
  }
  return(ceiling(k / 2))
}

# Random generation for the accum distribution
raccum <- function(nreps, k) {
  sapply(1:nreps, function(x) {
    total <- 0
    rolls <- 0
    while (total < k) {
      total <- total + sample(1:6, 1) + sample(1:6, 1)
      rolls <- rolls + 1
    }
    return(rolls)
  })
}
```