Hw5

Irlanda Ayon-Moreno

4/27/2018

```
check_sides <- function(x) {
  if (length(x) != 6) {
    stop("'sides' must be a vector of length 6")
  }
}

check_prob <- function(x) {
  if (sum(x) != 1) {
    stop("elements in 'prob' must add up to 1")
  }
}
```

```
die <- function(sides = 1:6, prob = rep(1/6,6)) {
  check_sides(sides)
  check_prob(prob)
  object <- list(sides=sides,prob=prob)
  class(object) <- "die"
  return(object)
}
```

```
die()
```

```
## $sides
## [1] 1 2 3 4 5 6
##
## $prob
## [1] 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667
##
## attr(,"class")
## [1] "die"
```

```
print.die <- function(die) {
  cat('object "die"\n\n')
  print.data.frame(data.frame(sides = die$sides, prob = die$prob))
  invisible(die)
}
```

```
die()
```

```
## object "die"
##
##   sides      prob
## 1     1 0.1666667
```

```
## 2     2 0.1666667
## 3     3 0.1666667
## 4     4 0.1666667
## 5     5 0.1666667
## 6     6 0.1666667
```

```
fair_die <- die()
fair_die
```

```
## object "die"
##
##   sides      prob
## 1     1 0.1666667
## 2     2 0.1666667
## 3     3 0.1666667
## 4     4 0.1666667
## 5     5 0.1666667
## 6     6 0.1666667
```

```
# die with non-standard sides
weird_die <- die(sides = c('i', 'ii', 'iii', 'iv', 'v', 'vi'))
weird_die
```

```
## object "die"
##
##   sides      prob
## 1     i 0.1666667
## 2    ii 0.1666667
## 3   iii 0.1666667
## 4    iv 0.1666667
## 5     v 0.1666667
## 6    vi 0.1666667
```

```
# create a loaded die
loaded_die <- die(prob = c(0.075, 0.1, 0.125, 0.15, 0.20, 0.35))
loaded_die
```

```
## object "die"
##
##   sides  prob
## 1     1 0.075
## 2     2 0.100
## 3     3 0.125
## 4     4 0.150
## 5     5 0.200
## 6     6 0.350
```

```
bad_die <- die(sides = c('a', 'b', 'c', 'd', 'e'))
```

```
## Error in check_sides(sides): 'sides' must be a vector of length 6
```

```r
# bad prob
bad_die <- die(
sides = c('a', 'b', 'c', 'd', 'e', 'f'),
prob = c(0.2, 0.1, 0.1, 0.1, 0.5, 0.1))
```

```
## Error in check_prob(prob): elements in 'prob' must add up to 1
```

```r
check_times <- function(x) {
  if (!is.double(x)) {
    stop("times must be a number")
  } else if ((x < 1) | (x %% 1 != 0)) {
    stop("times must be an integer greater than 0")
  }
}
```

```r
roll <- function(die, times = 1) {
  check_times(times)
  object <- list(rolls = sample(die$sides, prob = die$prob, size = times, replace = TRUE),
                 sides = die$sides,
                 prob = die$prob,
                 total = times)
  class(object) <- "roll"
  return(object)
}
```

```r
print.roll <- function(roll, ...) {
  cat('object "roll"\n\n')
  print(roll$rolls)
  invisible(roll)
}
```

```r
# roll fair die 50 times
fair_die <- die()
set.seed(123)
fair50 <- roll(fair_die, times = 50)
fair50
```

```
## object "roll"
##
##  [1] 3 6 4 1 1 2 5 1 5 4 1 4 6 5 2 1 3 2 3 1 1 6 5 1 5 6 5 5 3 2 1 1 6 6 2
## [36] 4 6 3 3 3 2 4 4 4 2 2 3 4 3 1
```

```r
# what's in fair50?
names(fair50)
```

```
## [1] "rolls" "sides" "prob"  "total"
```

```r
fair50$rolls
```

```
##  [1] 3 6 4 1 1 2 5 1 5 4 1 4 6 5 2 1 3 2 3 1 1 6 5 1 5 6 5 5 3 2 1 1 6 6 2
## [36] 4 6 3 3 3 2 4 4 4 2 2 3 4 3 1
```

```
fair50$sides
```

```
## [1] 1 2 3 4 5 6
```

```
fair50$prob
```

```
## [1] 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667 0.1666667
```

```
fair50$total
```

```
## [1] 50
```

```
# string die
str_die <- die(
  sides = c('a', 'b', 'c', 'd', 'e', 'f'),
  prob = c(0.075, 0.1, 0.125, 0.15, 0.20, 0.35)
)
# roll 20 times
set.seed(123)
str_rolls <- roll(str_die, times = 20)
names(str_rolls)
```

```
## [1] "rolls" "sides" "prob"  "total"
```

```
str_rolls
```

```
## object "roll"
##
##  [1] "f" "c" "e" "b" "a" "f" "e" "b" "d" "e" "a" "e" "d" "d" "f" "b" "f"
## [18] "f" "f" "a"
```

```
summary.roll <- function(roll, ...) {
  rolls <- roll$rolls
  side <- roll$sides
  count <- rep(0, 6)
  for (i in 1:6) {
    count[i] <- sum(side[i] == rolls)
  }
  prop <- count/roll$total
  frequencies <- data.frame(side, count, prop)
  object <- list(freqs = frequencies)
  class(object) <- "summary.roll"
  return(object)
}
```

```
print.summary.roll <- function(x, ...) {
  cat('summary "roll"\n\n')
  print(x$freqs)
  invisible(x)
}
```

```r
fair50_sum <- summary(fair50)
fair50_sum
```

```
## summary "roll"
##
##   side count prop
## 1    1    11 0.22
## 2    2     8 0.16
## 3    3     9 0.18
## 4    4     8 0.16
## 5    5     7 0.14
## 6    6     7 0.14
```

```r
class(fair50_sum)
```

```
## [1] "summary.roll"
```

```r
names(fair50_sum)
```

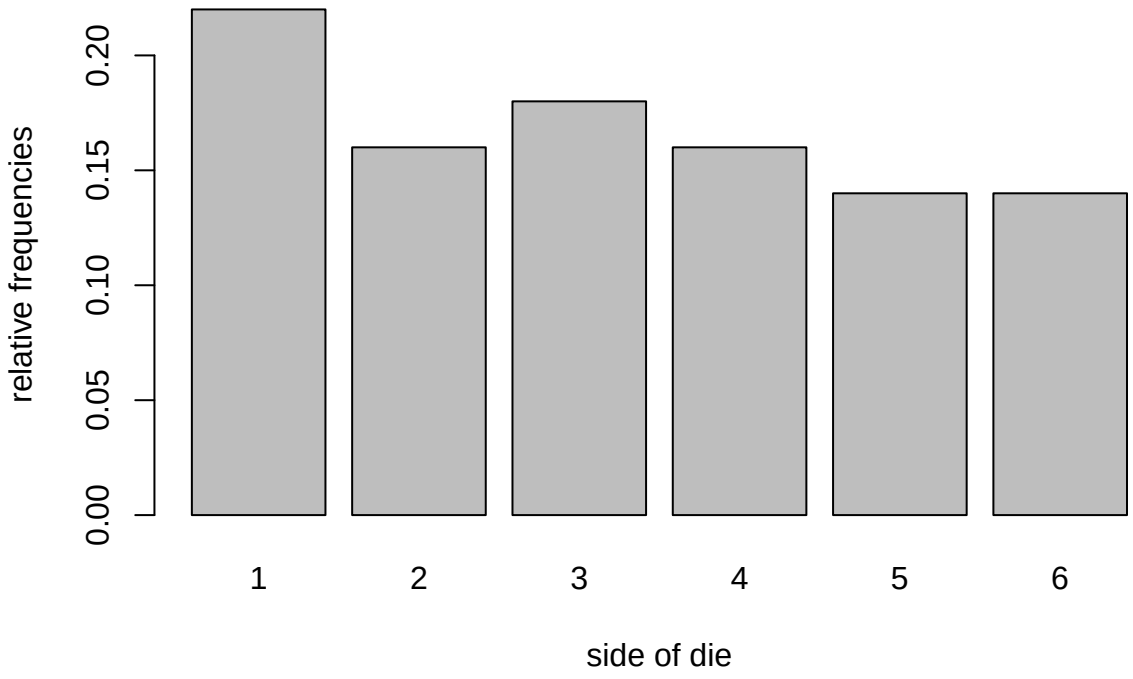
```
## [1] "freqs"
```

```r
fair50_sum$freqs
```

```
##   side count prop
## 1    1    11 0.22
## 2    2     8 0.16
## 3    3     9 0.18
## 4    4     8 0.16
## 5    5     7 0.14
## 6    6     7 0.14
```

```r
plot.roll <- function(roll, ...) {
  sum_r_e <- summary(roll)
  barplot(height = sum_r_e$freqs[["prop"]],
          names.arg = sum_r_e$freqs[["side"]],
          xlab = "side of die",
          ylab = "relative frequencies",
          main = (sprintf("Relative Frequencies in a series of %s coin tosses", roll$total))
  )
}
```

```r
plot(fair50)
```




```
make_roll <- function(die, rolls) {
  res <- list(
    rolls  = rolls,
    sides = die$sides,
    prob = die$prob,
    total = length(rolls))
  class(res) <- "roll"
  return(res)
}
```

```
"[.roll" <- function(x, i) {
  x$rolls[i]
}
```

```
"[<-.roll" <- function(x, i, value) {
  x$rolls[i] <- value
  make_roll(x, x$rolls)
}
```

```
"+.roll" <- function(x, incr) {
  if (length(incr) != 1 | incr <= 0) {
    stop("\ninvalid increament (must be positive)")
  }
  more_rolls <- roll(x, times = incr)
  make_roll(x, c(x$rolls, more_rolls$rolls))
}
```

```r
# roll fair die
set.seed(123)
fair_die <- die()
fair500 <- roll(fair_die, times = 500)
# summary method
summary(fair500)
```

```
## summary "roll"
##
##   side count  prop
## 1    1    80 0.160
## 2    2    81 0.162
## 3    3    92 0.184
## 4    4    92 0.184
## 5    5    72 0.144
## 6    6    83 0.166
```

```r
fair500[500]
```

```
## [1] 6
```

```r
fair500[500] <- 1
fair500[500]
```

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

```r
summary(fair500)
```

```
## summary "roll"
##
##   side count  prop
## 1    1    81 0.162
## 2    2    81 0.162
## 3    3    92 0.184
## 4    4    92 0.184
## 5    5    72 0.144
## 6    6    82 0.164
```

```r
fair600 <- fair500 + 100
summary(fair600)
```

```
## summary "roll"
##
##   side count      prop
## 1    1   100 0.1666667
## 2    2    97 0.1616667
## 3    3   104 0.1733333
## 4    4   109 0.1816667
## 5    5    91 0.1516667
## 6    6    99 0.1650000
```

```
plot(fair500, 500)
```