Programming: Practical 2b solutions
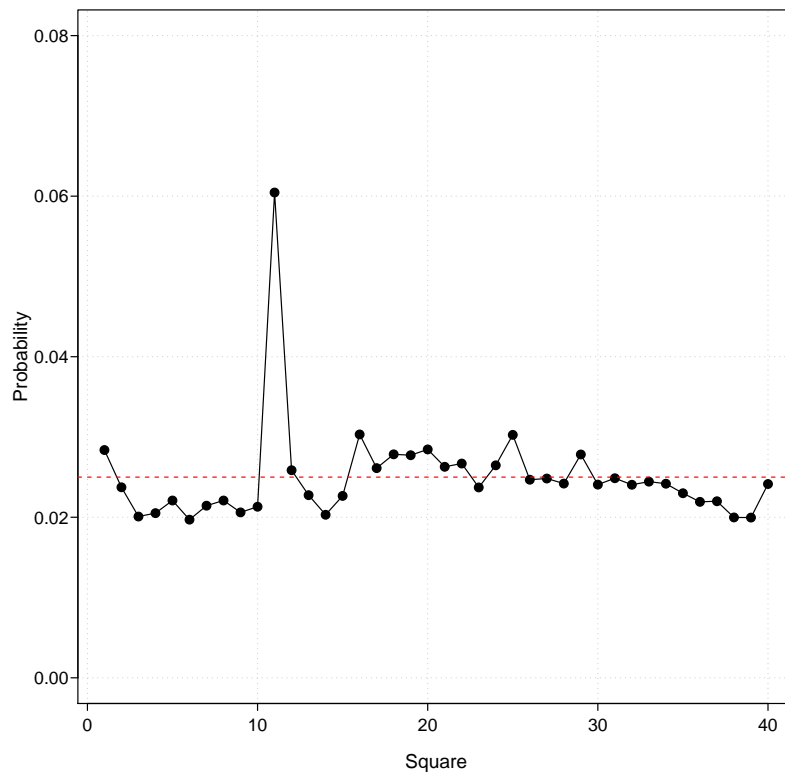

Figure 1: Distribution of resting places in a game of monopoly using 5×10^4 rolls.

```
##Dice function
RollTwoDiceWithDoubles = function(current) {

  df = data.frame(d1 = sample(1:6, 3, replace=TRUE),
    d2 = sample(1:6, 3, replace=TRUE))

  df$Total = apply(df, 1, sum)
  df$IsDouble = df$d1 == df$d2

  if(df$IsDouble[1] & df$IsDouble[2] & df$IsDouble[3]) {
    current = 11#Go To Jail
  } else if(df$IsDouble[1] & df$IsDouble[2]) {
    current = current + sum(df$Total[1:2])
  } else {
    current = current + df$Total[1]
  }
  return(current)
}


##Helper function to avoid code replication
CheckState = function(current) {
  if(current > 40) {
```

```r
    current = current - 40
  } else if(current < 1) {
    current = current + 40
  }
  return(current)
}
```

```r
UpdateStateVector = function(current, move, landings) {
  if(move != current){
    landings[current] = landings[current] + 1
  }
  return(landings)
}
```

```r
CommunityChest = function(current) {
  u = runif(1)
  goto = current#Default. Do nothing
  if(u < 1/16) {
    goto = 1#Go
  }else if(u < 2/16) {
    goto = 11#Jail
  }else if(u < 3/16) {
    goto = 2#Old Kent Rd
  }else if(u < 4/16) {
    goto = Chance(current)
  }
  return(goto)
}

Chance = function(current) {
  u = runif(1)
  goto = current#Default. Do nothing
  if(u < 1/16){
    goto = 1#Go
  }else if(u < 2/16){
    goto = 25#Tra Square
  }else if(u < 3/16){
    goto = 12#Pall Mall
  }else if(u < 4/16){
    goto = 11#Jail
  }else if(u < 5/16){
    goto = 16#Mary' Stat
  }else if(u < 6/16){
    goto = 40#Mayfair
  }else if(u < 7/16) {
    goto = CheckState(current - 3)#Must check, since goto maybe negative!
  }else if(u < 8/16){
```

```r
    if(current > 29  | current < 13){
      goto = 13
    }else {
      goto = 29
    }
  }
  return(goto)
}
```

```r
SimulateMonopoly = function(no_of_turns) {
  landings = numeric(40)
  ##Start GO
  current = 1
  for(i in 1:no_of_turns) {
    current = RollTwoDiceWithDoubles(current)
    current = CheckState(current)
    landings = UpdateStateVector(current, -1, landings)

    if(current == 3 | current == 18 | current == 34) {#Community Chest
      move = CommunityChest(current)
      landings = UpdateStateVector(move, current, landings)
      current = move
    } else if(current == 8 | current == 23 | current == 37) {#Chance
      move = Chance(current)
      landings = UpdateStateVector(move, current, landings)
      current = move
    }

    ##Go To Jail. Chance could also send you here by
    ##going back three places
    if(current == 31) {
      current = 11
      landings = UpdateStateVector(current, -1, landings)
    }

  }
  return(landings)
}
```

Neater Solutions

Some of the functions could be simplified.

```
## Rather than if statements, just work to mod 40
CheckState = function(current) {
  current = current %% 40
  if(current == 0)
    current = 40
  return(current)
}
```

```
## Use the sample command to move square
CommunityChest = function(current) {
  u = runif(1)
  goto = current#Default. Do nothing
  if(u < 3/16) {
    goto = sample(c(1, 11, 2), 1)
  } else if(u < 4/16) {
    goto = Chance(current)
  }
  return(goto)
}
```

```
## Use the sample command to move square
Chance = function(current) {
  u = runif(1)
  goto = current#Default. Do nothing
  if(u < 6/16){
    goto = sample(c(1, 25, 12, 11, 16, 50), 1)
  }else if(u < 7/16) {
    goto = CheckState(current - 3)#Must check, since goto maybe negative!
  }else if(u < 8/16){
    if(current > 29  | current < 13){
      goto = 13
    }else {
      goto = 29
    }
  }
  return(goto)
}
```