Practical 4

Complete this sheet as you work through it. Make sure you nicely indent your code.

1 Monopoly

1.1 Description of the problem

We wish to determine which properties a player is most likely to land on during a game of monopoly. To simplify things, we assume there is only a single player, we ignore everything to do with money, and we also ignore the 'Get out of Jail Free Cards'. The algorithm we will use is:

1. Begin the game on GO;
2. `current` := `current` + dice roll
3. Make a note of the new position.
   - If we land on 'Chance' or 'Community Chest', draw a card;
   - If we land on 'Go To Jail', move to Jail;
   - If we move, make a note of the new position;
4. Go back to step 2

After rolling the dice 100,000 times or so, stop.

1.2 Dice rolling

When we roll a single die, the probability mass function is:

$$\Pr[X = k] = 1/6 \quad \text{for } k = 1, \ldots, 6$$

i.e. all sides of the dice are equally likely. This means we can use the **sample** function to simulate a die roll:

```
sample(c(1, 2, 3, 4, 5, 6), 1)
#Or just the same
sample(seq(1, 6), 1)
#Or
sample(1:6, 1)
```

To roll two dice, we simply call this function:

Listing 1: Rolling two dice

```
RollTwoDice = function() {
  total = sample(1:6, 1) + sample(1:6, 1)
  return(total)
}
```

Complete: Call the function `RollTwoDice` four times, what values did you get? _____

Complete: What are the range of possible values `RollTwoDice` could return? _____

Complete: If we tried to simulate rolling two dice using **sample**`(2:12, 1)`, why would this be wrong?

_____

Complete: If we tried to simulate rolling two dice using `2*`**sample**`(1:6, 1)`, why would this be wrong?

_____

1.3 The Monopoly board

In Monopoly there are forty properties, or squares; see table 1 at the end of this practical for a complete list. If we set the first square 'Go' to be number 1, then we can represent all forty squares as a vector in **R**. For example

```
#This creates a vector of 40 values;
#All values are initially zero
landings = numeric(40)
```

Then, when we land on a square we simply increase the associated `landings` entry by one. Suppose we landed on 'Old Kent Rd', we would represent this as:

```
landings[2] = landings[2] + 1
```

since 'Old Kent Road' is square 2 (see table 1).

Complete: Write down the code if we landed on 'Free Parking'? _____

Complete: Write down the code if we landed on 'Mayfair'? _____

[Please turn over]

1.4 Going round the board

Our first go at simulating Monopoly will ignore the community chest, chance cards, and the 'Go To Jail' square. This means that we are simply going round the board. The code in listing 2 rolls the dice `no_of_rolls` times, and stores the squares that are landed on in the vector `landings`.

Listing 2: Basic Monopoly

```
1   SimulateMonopoly = function(no_of_rolls) {
2     landings = numeric(40)
3     #Start at GO
4     current = 1
5     for(i in 1:no_of_rolls) {
6       current = current + RollTwoDice()
7       if(current > 40) {
8         current = current - 40
9       }
10      landings[current] = landings[current] + 1
11    }
12      return(landings)
13  }
14  no_of_rolls = 50000
15  sim = SimulateMonopoly(no_of_rolls)
16  plot(sim/sum(sim), ylim=range(0.01, 0.04), type='l')
```

Complete: What does `landings = numeric(40)` do?_____

Complete: Explain the rationale for lines 9-12: _____

Complete: Why do we divide by `sum(sim)` above? _____

Complete: What happens to the plot as you increase `no_of_rolls`? _____

Complete: What values are the probabilities converging to (we will call this value the **baseline** probability)

_____

Complete: Add a horizontal line to your plot showing the **baseline** probability.

[Please turn over]

1.5 Incorporating Community Chest Cards

There are three community chest squares on the board - squares 3, 18 and 34. In the code below we will just consider square 3. There are sixteen cards in total, hence the probability of drawing any particular card is 1/16. In the code below we will **only implement the first two community chest cards**:

Listing 3: Basic Community Chest function

```
CommunityChest = function(current) {
  goto = current
  u = runif(1)
  if(u < 1/16) {
    goto = 1#Move to Go
  }else if(u < 2/16) {
    goto = 11#Go To Jail
  }
  return(goto)
}
```

This function takes in the `current` position, with probability 1/16 we 'Move to Go', with probability 1/16 we 'Go to Jail' and with probability 14/16 we stay in our current position. We now alter Listing 2 to incorporate the `communityChest` function:

Listing 4: Basic Monopoly

```
1  SimulateMonopoly1 = function(no_of_rolls) {
2    landings = numeric(40)
3    #Start at GO
4    current = 1
5    for(i in 1:no_of_rolls) {
6      current = current + RollTwoDice()
7      if(current > 40) {
8        current = current - 40
9      }
10     landings[current] = landings[current] + 1
11     if(current == 3) {
12       cc_move = CommunityChest(current)
13       if(cc_move != current){
14         current = cc_move
15         landings[current] = landings[current] + 1
16       }
17     }
18   }
19   return(landings)
20 }
21 no_of_rolls = 200000
22 sim2 = SimulateMonopoly1(no_of_rolls)
23 plot(sim2/sum(sim2), ylim=range(0.01, 0.04))
```

Complete: Explain the rationale for lines 14-20: _____

Complete: Add a horizontal line to your plot with the **baseline** probabilities.

2 Assignment

Each question adds an additional layer of complexity to your code.

1. Run the simulation function `SimulateMonopoly1`. [14%]
2. Add in the two other community squares, i.e. squares 18 and 34 into the `SimulateMonopoly1` code; [+7%]
3. Add in 'Go to Old Kent Road' into your `CommunityChest` function;[+7%]
4. Square 31 is 'Go To Jail.' Implement this in your main simulation function;[+7%]
5. Create a `Chance` function, that implements the first six Chance cards. When you land on a Chance square, call this function;[+10%]
6. Add in Community Chest card four;[+10%]
7. Add in Chance card 8.[+10%]
8. Add in Chance card 7, 'Go back 3 spaces'.[+10%]
9. Rolling a double (a pair of 1's, 2's, ..., 6's) is special: [+10%]
   a) Roll two dice (`total1`): `total_score = total1`
   b) If you get a double, roll again (`total2`) and `total_score = total1 + total2`
   c) If you get a double, roll again (`total3`) and `total_score = total1 + total2 + total3`
   d) If roll three is a double, Go To Jail, otherwise move `total_score`

Other marks are awarded for code presentation and style [+15%]

What to hand in

a A front page as described in Practical 2.

b Submit your nicely formated code. This can printed directly from `RStudio` or from Word. If you use Word, make sure the formatting isn't messed up.
   - There should only be a single `SimulateMonopoly1` function.
   - Of course, there will be other functions. For example `Chance`, `CommunityChest`.
   - If you submit more than one `SimulateMonopoly1`, marks will be deducted.
   - If you submit a function that **produces** an error, i.e. it doesn't even run without crashing, you may get zero marks.
   - Don't submit your plot commands. Just submit your R functions.

c Plot the probability of being on a certain square, against property based on your code. Add in a **dashed** horizontal line with the **baseline** probabilities.

d State clearly how many simulations you used. When working through the practical use about 10,000 simulations, but your final results should probably be at least 100,000 or even more depending on your computer speed.

Remember to label all axis and put your name & student id on your work. Also, put your student number on all plots.

3 Additional Information

Community Chest Cards

There are three community chest areas on the board (see Table 1). In total, there are 16 community chest cards.

1. Advance to Go;
2. Go to jail;
3. Go to Old Kent Road;
4. Take a Chance card instead;

Chance Cards

A Chance card is most likely to move players. There are three chance areas on the board (see Table 1). There are 16 chance cards in total, of which eight cards move the player:

1. Advance to Go;
2. Advance to Trafalgar Square;
3. Advance to Pall Mall;
4. Go directly to Jail;
5. Take a trip to Marylebone Station
6. Advance to Mayfair
7. Go back 3 spaces;
8. Advance token to nearest Utility. The utility squares are the water works and the electric company.

Square Number	Name	Square Number	Name
1	Go	11	Jail
2	Old Kent Road	12	Pall Mall
3	Community Chest	13	Electric Company
4	WhiteChapel Road	14	Whitehall
5	Income tax	15	Northumberland Avenue
6	King's Cross Station	16	Marylebone station
7	The Angel Islington	17	Bow Street
8	Chance	18	Community Chest
9	Euston Road	19	Marlborough Street
10	Pentonville Road	20	Vine Street
21	Free Parking	31	Go To Jail
22	Strand	32	Regent Street
23	Chance	33	Oxford Street
24	Fleet Street	34	Community Chest
25	Trafalgar Square	35	Bond Street
26	Fenchurch Street Station	36	Liverpool St Station
27	Leicester Square	37	Chance
28	Coventry St	38	Park Lane
29	Water Works	39	Super Tax
30	Piccadilly	40	Mayfair

Table 1: *Monopoly squares with associated square numbers*