

Monopoly: Some hints

• Debugging

- For question 2, do each part (a)–(h) one at a time, checking the graph at each stage...
- ...then, if you get something weird, you'll have more idea where you went wrong
- Each time you edit your code, do so in a different script, so you don't screw up the code that you know works!
- Indent your code – the Rstudio text editor usually does this automatically – indenting makes your code more structured and easier to spot where things might be wrong
- Check your brackets, and make sure each thing you add to the main code is in the right 'section'

Monopoly: Some hints

• Debugging

- For question 2, do each part (a)–(h) one at a time, checking the graph at each stage...
- ...then, if you get something weird, you'll have more idea where you went wrong
- Each time you edit your code, do so in a different script, so you don't screw up the code that you know works!
- Indent your code – the Rstudio text editor usually does this automatically – indenting makes your code more structured and easier to spot where things might be wrong
- Check your brackets, and make sure each thing you add to the main code is in the right 'section'

Monopoly: Some hints

• Debugging

- For question 2, do each part (a)–(h) one at a time, checking the graph at each stage...
- ...then, if you get something weird, you'll have more idea where you went wrong
- Each time you edit your code, do so in a different script, so you don't screw up the code that you know works!
- Indent your code – the Rstudio text editor usually does this automatically – indenting makes your code more structured and easier to spot where things might be wrong
- Check your brackets, and make sure each thing you add to the main code is in the right 'section'

Monopoly: Some hints

• Debugging

- For question 2, do each part (a)–(h) one at a time, checking the graph at each stage...
- ...then, if you get something weird, you'll have more idea where you went wrong
- Each time you edit your code, do so in a different script, so you don't screw up the code that you know works!
- Indent your code – the Rstudio text editor usually does this automatically – indenting makes your code more structured and easier to spot where things might be wrong
- Check your brackets, and make sure each thing you add to the main code is in the right 'section'

Monopoly: Some hints

• Debugging

- For question 2, do each part (a)–(h) one at a time, checking the graph at each stage...
- ...then, if you get something weird, you'll have more idea where you went wrong
- Each time you edit your code, do so in a different script, so you don't screw up the code that you know works!
- Indent your code – the Rstudio text editor usually does this automatically – indenting makes your code more structured and easier to spot where things might be wrong
- Check your brackets, and make sure each thing you add to the main code is in the right 'section'

Monopoly: Some hints

• Debugging

- For question 2, do each part (a)–(h) one at a time, checking the graph at each stage...
- ...then, if you get something weird, you'll have more idea where you went wrong
- Each time you edit your code, do so in a different script, so you don't screw up the code that you know works!
- Indent your code – the Rstudio text editor usually does this automatically – indenting makes your code more structured and easier to spot where things might be wrong
- Check your brackets, and make sure each thing you add to the main code is in the right 'section'

Monopoly: Some hints

- **What to look for**

- Initially, all probabilities should hover around $1/40$
- Once you start adding in extra layers (chance, community chest etc) this will change for some squares
- Lots of things send you to Jail! So expect a spike here...

- **Rolling doubles – this bit is hard!**

- Unless you're confident your code for part (h) works, leave it out!
- Including a function for this which doesn't work properly could cost you dearly!

Monopoly: Some hints

- **What to look for**

- Initially, all probabilities should hover around $1/40$
- Once you start adding in extra layers (chance, community chest etc) this will change for some squares
- Lots of things send you to Jail! So expect a spike here...

- **Rolling doubles – this bit is hard!**

- Unless you're confident your code for part (h) works, leave it out!
- Including a function for this which doesn't work properly could cost you dearly!

Monopoly: Some hints

● What to look for

- Initially, all probabilities should hover around $1/40$
- Once you start adding in extra layers (chance, community chest etc) this will change for some squares
- Lots of things send you to Jail! So expect a spike here...

● Rolling doubles – this bit is hard!

- Unless you're confident your code for part (h) works, leave it out!
- Including a function for this which doesn't work properly could cost you dearly!

Monopoly: Some hints

● What to look for

- Initially, all probabilities should hover around $1/40$
- Once you start adding in extra layers (chance, community chest etc) this will change for some squares
- Lots of things send you to Jail! So expect a spike here...

● Rolling doubles – this bit is hard!

- Unless you're confident your code for part (h) works, leave it out!
- Including a function for this which doesn't work properly could cost you dearly!

Monopoly: Some hints

● What to look for

- Initially, all probabilities should hover around $1/40$
- Once you start adding in extra layers (chance, community chest etc) this will change for some squares
- Lots of things send you to Jail! So expect a spike here...

● Rolling doubles – this bit is hard!

- Unless you're confident your code for part (h) works, leave it out!
- Including a function for this which doesn't work properly could cost you dearly!

Monopoly: Some hints

- **What to look for**

- Initially, all probabilities should hover around $1/40$
- Once you start adding in extra layers (chance, community chest etc) this will change for some squares
- Lots of things send you to Jail! So expect a spike here...

- **Rolling doubles – this bit is hard!**

- Unless you're confident your code for part (h) works, leave it out!
- Including a function for this which doesn't work properly could cost you dearly!

Monopoly: Some hints

- **What to look for**

- Initially, all probabilities should hover around $1/40$
- Once you start adding in extra layers (chance, community chest etc) this will change for some squares
- Lots of things send you to Jail! So expect a spike here...

- **Rolling doubles – this bit is hard!**

- Unless you're confident your code for part (h) works, leave it out!
- Including a function for this which doesn't work properly could cost you dearly!

Monopoly: Some hints

- **What to look for**

- Initially, all probabilities should hover around $1/40$
- Once you start adding in extra layers (chance, community chest etc) this will change for some squares
- Lots of things send you to Jail! So expect a spike here...

- **Rolling doubles – this bit is hard!**

- Unless you're confident your code for part (h) works, leave it out!
- Including a function for this which doesn't work properly could cost you dearly!

Rolling doubles

```
> 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
> df
```

	d1	d2	Total	IsDouble
1	6	3	9	FALSE
2	4	3	7	FALSE
3	2	2	4	TRUE

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

Rolling doubles

```
> 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
> df
```

	d1	d2	Total	IsDouble
1	6	3	9	FALSE
2	4	3	7	FALSE
3	2	2	4	TRUE

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

Sticker album: Some hints

Ultimate goals:

- Find out how many packets of stickers are needed to fill the album!
- How much will this cost?
- Histograms and other numerical summaries...

Sticker album: Some hints

Ultimate goals:

- Find out how many packets of stickers are needed to fill the album!
- How much will this cost?
- Histograms and other numerical summaries...

Sticker album: Some hints

Ultimate goals:

- Find out how many packets of stickers are needed to fill the album!
- How much will this cost?
- Histograms and other numerical summaries...