Advent of Code 2020

Zach Bogart

Day 5: Binary Boarding

https://adventofcode.com/2020/day/5

```
input = tibble(x = read_lines("inputs/05-input.txt"))
```

Part 1

- make a function to narrow in on the range of possible values
- find the one with the highest id

```
# go through instructions, making range of options smaller and smaller
narrow_in_on_location = function(instructions, options, front_letter) {

  current_options = options

  for (letter in str_split(instructions, "")[[1]]) {
    midpoint = length(current_options) / 2
    if (letter == front_letter) {
      current_options = current_options[1:midpoint]
    } else {
      current_options = current_options[(midpoint + 1): length(current_options)]
    }
  }

  return(current_options)
}

# make two columns, for row and column
part1 = input %>%
  rowwise() %>%
  mutate(row = narrow_in_on_location(str_sub(x, 1, 7), seq(0,127, 1), "F"),
         column = narrow_in_on_location(str_sub(x, 8, 10), seq(0, 7, 1), "L"),
         seat_id = row * 8 + column) %>%
  arrange(desc(seat_id))

head(part1)
```

```
## # A tibble: 6 x 4
## # Rowwise:
##   x            row column seat_id
##   <chr>      <dbl>  <dbl>   <dbl>
## 1 BBBBBFFLRL   124      2     994
## 2 BBBBBFFLLR   124      1     993
## 3 BBBBBFFLLL   124      0     992
## 4 BBBBFBBRRR   123      7     991
## 5 BBBBFBBRRL   123      6     990
## 6 BBBBFBBRLR   123      5     989
```

```
answer = part1[1, "seat_id"]
answer
```

```
## # A tibble: 1 x 1
## # Rowwise:
##   seat_id
##     <dbl>
## 1     994
```

Part 2

- Compare full list of seats with just the filled ones
- find the odd one out

```
filled_seats = as.vector(part1$seat_id)
all_seats = seq(
  head(filled_seats, n=1),
  tail(filled_seats, n=1)
)
setdiff(all_seats, filled_seats)
```

```
## [1] 741
```