MET CS544

Weilin Lu

Assignment 2

PART 1

Age	Number of people	% of total	BMI of above 30	Probability
18-34	4250	42.5%	1062	24.99%
35-49	2850	28.5%	1710	60%
50-64	1640	16.4%	656	40%
Over 65	1260	12.6%	189	15%

a:)

P(have BMI of above 30)=(0.425*0.2499+0.6*0.285+0.4*0.164+0.15*0.126)*100

=36.17075%

b:)

P(18-34 have BMI of above 30)=[(0.2499*0.425)/0.3617075]*100

=29.63%(rounded)

c:)

P(35-49 have BMI of above 30)=[(0.6*0.285)/0.3617075]*100

=47.28%(rounded)

d:)

P(50-64 have BMI of above 30)=[(0.4*0.164)/0.3617075]*100

=18.14%(rounded)

e:)

P(Over 65 have BMI of above 30)=[(0.15*0.126)/0.3617075]*100

=5.23%(rounded)

PART 2

CODE AND OUPTPUT

a:)

```
#part2
print("Part2")
output <- rolldie(3)
output$probs <- 1/nrow(output)

#a
print("a")
sumRow <- rowSums(output[c('X1', 'X2', 'X3')])
Three_Eight <- output[sumRow >3 & sumRow <8,]
print(Three_Eight)
print(sum(Three_Eight$probs))
print("=============================================================================================")
```

```
Attaching package: 'prob'

The following objects are masked from 'package:base':

    intersect, setdiff, union

[1] "Part2"
[1] "a"
    X1 X2 X3      probs
2    2  1  1 0.00462963
3    3  1  1 0.00462963
4    4  1  1 0.00462963
5    5  1  1 0.00462963
7    1  2  1 0.00462963
8    2  2  1 0.00462963
9    3  2  1 0.00462963
10   4  2  1 0.00462963
13   1  3  1 0.00462963
14   2  3  1 0.00462963
15   3  3  1 0.00462963
19   1  4  1 0.00462963
20   2  4  1 0.00462963
25   1  5  1 0.00462963
37   1  1  2 0.00462963
38   2  1  2 0.00462963
39   3  1  2 0.00462963
40   4  1  2 0.00462963
43   1  2  2 0.00462963
44   2  2  2 0.00462963
45   3  2  2 0.00462963
49   1  3  2 0.00462963
50   2  3  2 0.00462963
55   1  4  2 0.00462963
73   1  1  3 0.00462963
74   2  1  3 0.00462963
75   3  1  3 0.00462963
79   1  2  3 0.00462963
80   2  2  3 0.00462963
85   1  3  3 0.00462963
109  1  1  4 0.00462963
110  2  1  4 0.00462963
115  1  2  4 0.00462963
145  1  1  5 0.00462963
[1] 0.1574074
[1] "========================================================================================="
```

b:)

```
#b
print("b")
ThreeRow <- (output['X1'] == output['X2']) &(output['X2'] == output['X3'])
all_three <- output[ThreeRow,]
print(all_three)
print(sum(all_three$probs))
print("=========================================================================================")
```

```
[1] "b"
    X1 X2 X3      probs
1    1  1  1 0.00462963
44   2  2  2 0.00462963
87   3  3  3 0.00462963
130  4  4  4 0.00462963
173  5  5  5 0.00462963
216  6  6  6 0.00462963
[1] 0.02777778
[1] "========================================================================================="
```

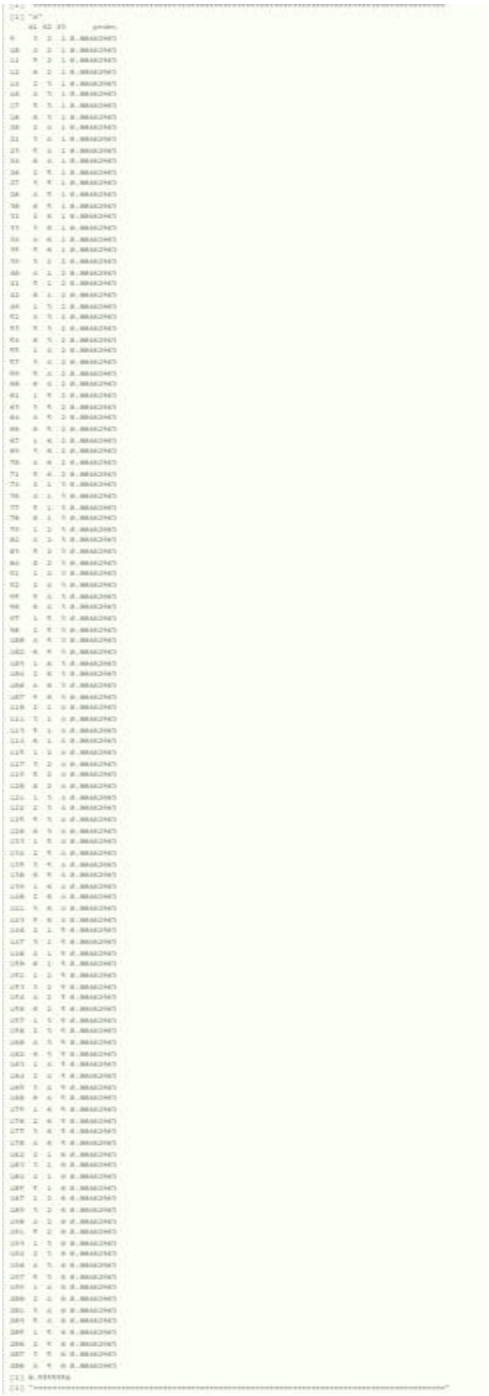
c:)

```
#c
print("c")
twoRow <- (output['X1'] == output['X2']) | (output['X1'] == output['X3']) |(output['X2'] == output['X3'])
TwoRow <- twoRow &(!ThreeRow)
Two <- output[TwoRow,]
print(Two)
print(sum(Two$probs))
print("=========================================================================================")
```

```
[1] "c"
    X1 X2 X3      profit
2    2  1  1 0.004629630
3    3  1  1 0.004629630
4    4  1  1 0.004629630
5    5  1  1 0.004629630
6    6  1  1 0.004629630
7    1  2  1 0.004629630
8    2  2  1 0.004629630
13   1  3  1 0.004629630
15   3  3  1 0.004629630
19   1  4  1 0.004629630
22   4  4  1 0.004629630
25   1  5  1 0.004629630
29   5  5  1 0.004629630
31   1  6  1 0.004629630
36   6  6  1 0.004629630
37   1  1  2 0.004629630
38   2  1  2 0.004629630
43   1  3  2 0.004629630
45   3  3  2 0.004629630
46   4  3  2 0.004629630
47   5  3  2 0.004629630
48   6  3  2 0.004629630
50   2  3  2 0.004629630
51   3  3  2 0.004629630
56   2  4  2 0.004629630
58   4  4  2 0.004629630
62   2  5  2 0.004629630
65   5  5  2 0.004629630
68   2  6  2 0.004629630
72   6  6  2 0.004629630
73   1  1  3 0.004629630
75   3  1  3 0.004629630
80   2  2  3 0.004629630
81   3  2  3 0.004629630
85   1  3  3 0.004629630
86   2  3  3 0.004629630
88   4  3  3 0.004629630
89   5  3  3 0.004629630
90   6  3  3 0.004629630
93   3  4  3 0.004629630
94   4  4  3 0.004629630
99   3  5  3 0.004629630
101  5  5  3 0.004629630
105  3  6  3 0.004629630
108  6  6  3 0.004629630
109  1  1  4 0.004629630
112  4  1  4 0.004629630
116  2  2  4 0.004629630
118  4  2  4 0.004629630
123  3  3  4 0.004629630
124  4  3  4 0.004629630
127  1  4  4 0.004629630
128  2  4  4 0.004629630
129  3  4  4 0.004629630
131  5  4  4 0.004629630
132  6  4  4 0.004629630
136  4  5  4 0.004629630
137  5  5  4 0.004629630
142  4  6  4 0.004629630
144  6  6  4 0.004629630
145  1  1  5 0.004629630
149  5  1  5 0.004629630
152  2  2  5 0.004629630
155  5  2  5 0.004629630
159  3  3  5 0.004629630
161  5  3  5 0.004629630
166  4  4  5 0.004629630
167  5  4  5 0.004629630
169  1  5  5 0.004629630
170  2  5  5 0.004629630
171  3  5  5 0.004629630
173  4  5  5 0.004629630
174  6  5  5 0.004629630
179  5  6  5 0.004629630
180  6  6  5 0.004629630
181  1  1  6 0.004629630
186  6  1  6 0.004629630
188  2  3  6 0.004629630
192  6  3  6 0.004629630
195  3  3  6 0.004629630
198  6  3  6 0.004629630
202  4  4  6 0.004629630
204  6  4  6 0.004629630
209  5  5  6 0.004629630
210  6  5  6 0.004629630
211  1  6  6 0.004629630
212  2  6  6 0.004629630
213  3  6  6 0.004629630
214  4  6  6 0.004629630
215  5  6  6 0.004629630
[1] 0.4166667
[1] "================================================================================"
```

d:)

```
#d
print("d")
None <- (output['X1'] != output['X2']) & (output['X1'] != output['X3']) & (output['X2'] != output['X3'])
NoneIdentical <- output[None,]
print(NoneIdentical)
print(sum(NoneIdentical$probs))
print("=============================================================================================")
```



e:)

```
#e
print("e")
Two_identical <- twoRow &(!ThreeRow) & output[sumRow >3 & sumRow <8,]
two_indentical <- output[Two_identical,]
print(two_indentical)
print(sum(two_indentical$probs))
print("=============================================================================================")
```

```
[1] "e"
    X1 X2 X3      probs
2    2  1  1 0.00462963
3    3  1  1 0.00462963
4    4  1  1 0.00462963
5    5  1  1 0.00462963
6    6  1  1 0.00462963
7    1  2  1 0.00462963
8    2  2  1 0.00462963
13   1  3  1 0.00462963
15   3  3  1 0.00462963
19   1  4  1 0.00462963
22   4  4  1 0.00462963
25   1  5  1 0.00462963
29   5  5  1 0.00462963
31   1  6  1 0.00462963
36   6  6  1 0.00462963
37   1  1  2 0.00462963
38   2  1  2 0.00462963
43   1  2  2 0.00462963
45   3  2  2 0.00462963
46   4  2  2 0.00462963
47   5  2  2 0.00462963
48   6  2  2 0.00462963
50   2  3  2 0.00462963
51   3  3  2 0.00462963
56   2  4  2 0.00462963
58   4  4  2 0.00462963
62   2  5  2 0.00462963
65   5  5  2 0.00462963
68   2  6  2 0.00462963
72   6  6  2 0.00462963
73   1  1  3 0.00462963
75   3  1  3 0.00462963
80   2  2  3 0.00462963
81   3  2  3 0.00462963
85   1  3  3 0.00462963
86   2  3  3 0.00462963
88   4  3  3 0.00462963
89   5  3  3 0.00462963
90   6  3  3 0.00462963
93   3  4  3 0.00462963
94   4  4  3 0.00462963
99   3  5  3 0.00462963
101  5  5  3 0.00462963
105  3  6  3 0.00462963
108  6  6  3 0.00462963
109  1  1  4 0.00462963
112  4  1  4 0.00462963
116  2  2  4 0.00462963
118  4  2  4 0.00462963
123  3  3  4 0.00462963
124  4  3  4 0.00462963
127  1  4  4 0.00462963
128  2  4  4 0.00462963
129  3  4  4 0.00462963
131  5  4  4 0.00462963
132  6  4  4 0.00462963
136  4  5  4 0.00462963
138  6  5  4 0.00462963
139  1  6  4 0.00462963
140  2  6  4 0.00462963
141  3  6  4 0.00462963
142  4  6  4 0.00462963
143  5  6  4 0.00462963
144  6  6  4 0.00462963
149  5  1  5 0.00462963
151  1  2  5 0.00462963
155  5  2  5 0.00462963
158  2  3  5 0.00462963
161  5  3  5 0.00462963
165  3  4  5 0.00462963
167  5  4  5 0.00462963
172  4  5  5 0.00462963
173  5  5  5 0.00462963
174  6  5  5 0.00462963
179  5  6  5 0.00462963
181  1  1  6 0.00462963
182  2  1  6 0.00462963
183  3  1  6 0.00462963
184  4  1  6 0.00462963
186  6  1  6 0.00462963
187  1  2  6 0.00462963
192  6  2  6 0.00462963
194  2  3  6 0.00462963
198  6  3  6 0.00462963
201  3  4  6 0.00462963
204  6  4  6 0.00462963
208  4  5  6 0.00462963
209  5  5  6 0.00462963
211  1  6  6 0.00462963
216  6  6  6 0.00462963
[1] 0.4166667
[1] "------------------------------------------------------------------------------------------"
```

PART 3

CODE AND OUPUT

```r
sum_of_first_N_even_squares <- function(n){
    n = n
    sum = 0
    for (i in 1:n){
        n <- 2*i
        sum <- sum + n^2
    }
    print(sum)
}
print("Sum of first 2 even function")
sum_of_first_N_even_squares(2)
print("Sum of first 5 even function")
sum_of_first_N_even_squares(5)
print("Sum of first 10 even function")
sum_of_first_N_even_squares(10)
```

```
[1] "Sum of first 2 even function"
[1] 20
[1] "Sum of first 5 even function"
[1] 220
[1] "Sum of first 10 even function"
[1] 1540
```

PART 4

a:)

```r
#PART 4
print("PART 4")
tsla <- read.csv("https://people.bu.edu/kalathur/datasets/TSLA.csv")

#a
print("a")
close <- summary(tsla$Close)
names(close) <- c("Min","Q1","Q2","Mean","Q3","Max")
print(close)
print("==========================================================================================")
```

```
[1] "a"
  Min    Q1    Q2  Mean    Q3   Max
109.0 225.0 272.0 263.1 302.5 400.0
[1] "=========================================================================================="
```

b:)

```r
#b
print("b")
minP <- which.min(tsla$Close)
tc <- tsla$Close[minP]
td <- tsla$Date[minP]
print(sprintf("The minimum Tesla value of %d is at row %d on %s", tc, minP, td))
print("=============================================================================================")
```

```
[1] "b"
[1] "The minimum Tesla value of 109 is at row 248 on 12/27/22"
[1] "============================================================================================="
```

c:)

```r
#c
print("c")
maxP <- which.max(tsla$Close)
tc <- tsla$Close[maxP]
td <- tsla$Date[maxP]
print(sprintf("The maximum Tesla value of %d is at row %d on %s", tc, maxP, td))
print("=============================================================================================")
```

```
[1] "c"
[1] "The maximum Tesla value of 400 is at row 1 on 1/3/22"
[1] "============================================================================================="
```

d:)

```r
#d
print("d")
Profit <- sum((tsla$Close - tsla$Open) > 0, na.rm = TRUE)
Prob <- Profit / totalRow
Percentage <- Prob*100
print(sprintf("The probability is %f percent ",Percentage))
print("=============================================================================================")
```

```
[1] "d"
[1] "The probability is 45.019920 percent "
[1] "============================================================================================="
```

e:)

```r
#e
print("e")
One_Million <- sum(tsla$Volume > 100000000, na.rm = TRUE)
Prob <- One_Million / totalRow
Percentage <- Prob*100
print(sprintf("The probability is %f percent ",Percentage))
print("=============================================================================================")
```

```
[1] "e"
[1] "The probability is 22.310757 percent "
[1] "============================================================================================="
```

f:)

```r
#f
print("f")
Profit<- sum((tsla$Close - tsla$Open) > 0 & tsla$Volume > 100000000, na.rm = TRUE)
Prob <- Profit / totalRow
Percentage <- Prob*100
print(sprintf("The probability is %f percent ",Percentage))
print("=============================================================================================")
```

```
[1] "f"
[1] "The probability is 10.358566 percent "
[1] "============================================================================================="
```

g:)

```r
#g
print("g")
Sell_out<- tail(tsla$Close,1)
Buy_in <- which.min(tsla$Close)
profit <- Sell_out - buyinPrice
print(sprintf("Buy in with the lowest price $%d, and sold out :$%d on the last day. The net profit is :$%d",
```

```
[1] "g"
[1] "Buy in with the lowest price $248, and sold out :$123 on the last day. The net profit is :$-125"
```