CS544 Homework 2

Part 1

Age	Number of People	Rate of people in Surveyed	Number of BMI above 30	Probability of BMI above 30
18-34	4250	0.425	1062	0.24988
35-49	2850	0.285	1710	0.6
50-64	1640	0.164	656	0.4
>= 65	1260	0.126	189	0.15

a:

P(selected person in survey have BMI above 30)

$= 0.425 \times 0.24988 + 0.285 \times 0.6 + 0.164 \times 0.4 + 0.126 \times 0.15$

$= 0.106199 + 0.171 + 0.0656 + 0.0189$

$= 0.3617(\text{rounded})$

Beyes Rule: $P(B|A) = \frac{P(A|B)\times P(B)}{P(A)}$

b:

P(person who is in age 18-34 have BMI above 30)

$= \frac{(0.24988 \times 0.425)}{0.3617}$

=0.2936(rounded)

c:

P(person who is in age 35-49 have BMI above 30)

$= \frac{(0.6\times 0.285)}{0.3617}$

=0.4728(rounded)

d:

P(person who is in age 50-64 have BMI above 30)

$= \frac{(0.4\times 0.164)}{0.3617}$

=0.1814(rounded)

e:

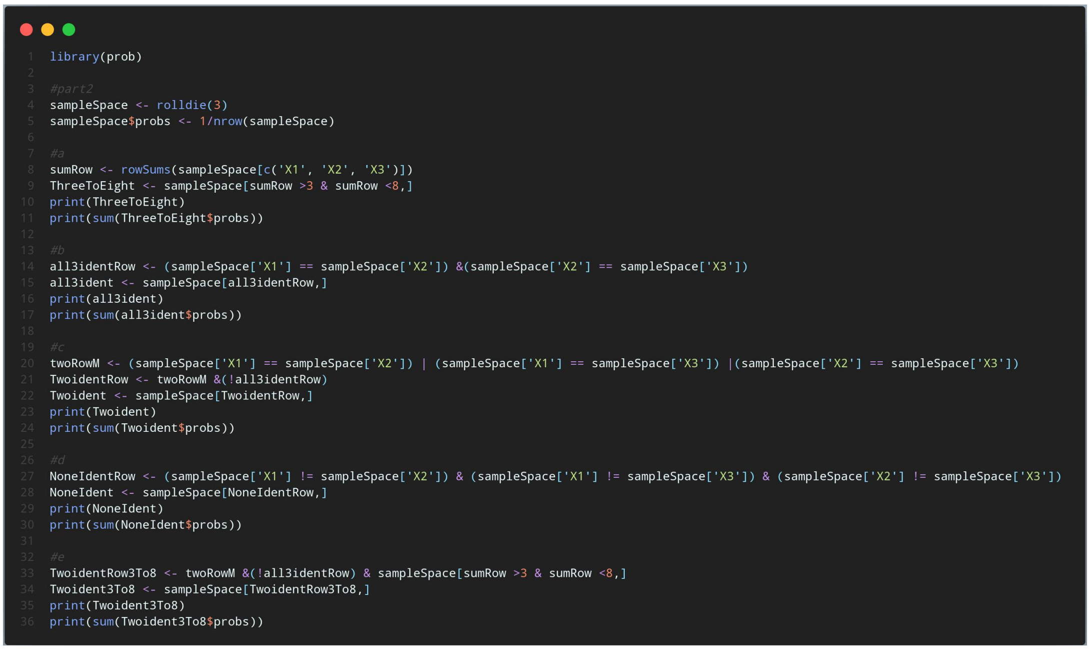
P(person who is in age >=65 have BMI above 30)

$$=\frac{(0.15\times 0.126)}{0.3617}$$

=0.0523(rounded)

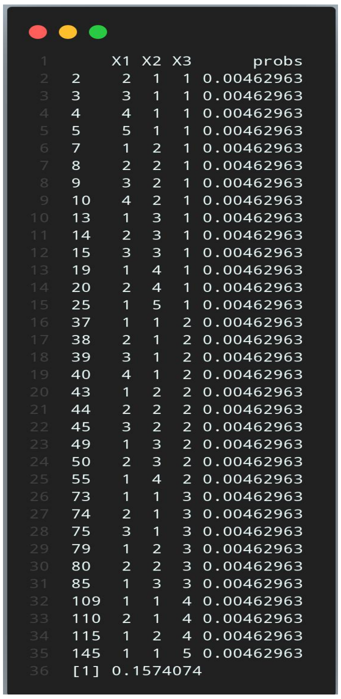
Part 2

code



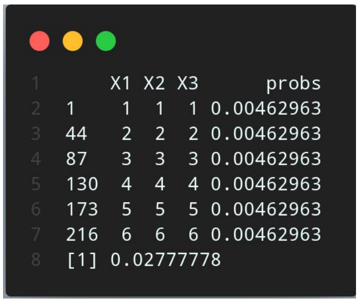
```
library(prob)

#part2
sampleSpace <- rolldie(3)
sampleSpace$probs <- 1/nrow(sampleSpace)

#a
sumRow <- rowSums(sampleSpace[c('X1', 'X2', 'X3')])
ThreeToEight <- sampleSpace[sumRow >3 & sumRow <8,]
print(ThreeToEight)
print(sum(ThreeToEight$probs))

#b
all3identRow <- (sampleSpace['X1'] == sampleSpace['X2']) &(sampleSpace['X2'] == sampleSpace['X3'])
all3ident <- sampleSpace[all3identRow,]
print(all3ident)
print(sum(all3ident$probs))

#c
twoRowM <- (sampleSpace['X1'] == sampleSpace['X2']) | (sampleSpace['X1'] == sampleSpace['X3']) |(sampleSpace['X2'] == sampleSpace['X3'])
TwoidentRow <- twoRowM &(!all3identRow)
Twoident <- sampleSpace[TwoidentRow,]
print(Twoident)
print(sum(Twoident$probs))

#d
NoneIdentRow <- (sampleSpace['X1'] != sampleSpace['X2']) & (sampleSpace['X1'] != sampleSpace['X3']) & (sampleSpace['X2'] != sampleSpace['X3'])
NoneIdent <- sampleSpace[NoneIdentRow,]
print(NoneIdent)
print(sum(NoneIdent$probs))

#e
TwoidentRow3To8 <- twoRowM &(!all3identRow) & sampleSpace[sumRow >3 & sumRow <8,]
Twoident3To8 <- sampleSpace[TwoidentRow3To8,]
print(Twoident3To8)
print(sum(Twoident3To8$probs))
```

a:output



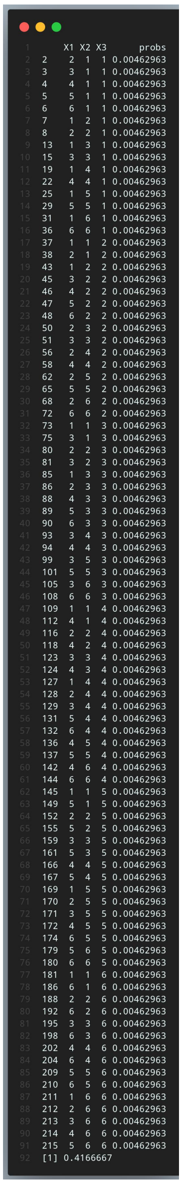
```
      X1 X2 X3      probs
2      2  1  1 0.00462963
3      3  1  1 0.00462963
4      4  1  1 0.00462963
5      5  1  1 0.00462963
7      1  2  1 0.00462963
8      2  2  1 0.00462963
9      3  2  1 0.00462963
10     4  2  1 0.00462963
13     1  3  1 0.00462963
14     2  3  1 0.00462963
15     3  3  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
```

b:output



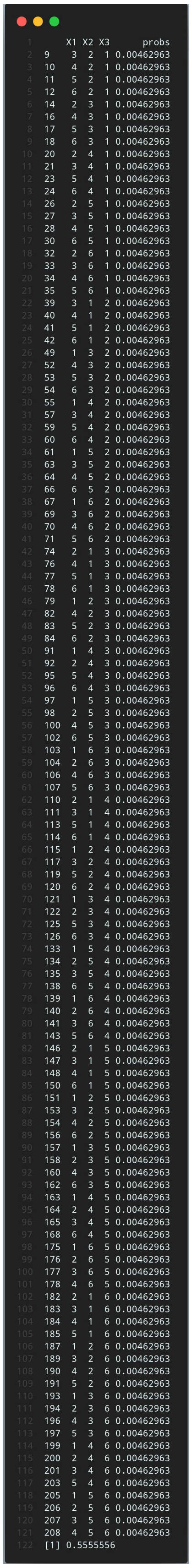
```
    X1 X2 X3      probs
1    1  1  1 0.00462963
44   2  2  2 0.00462963
87   3  3  3 0.00462963
130  4  4  4 0.00462963
173  5  5  5 0.00462963
216  6  6  6 0.00462963
[1] 0.02777778
```

c:output



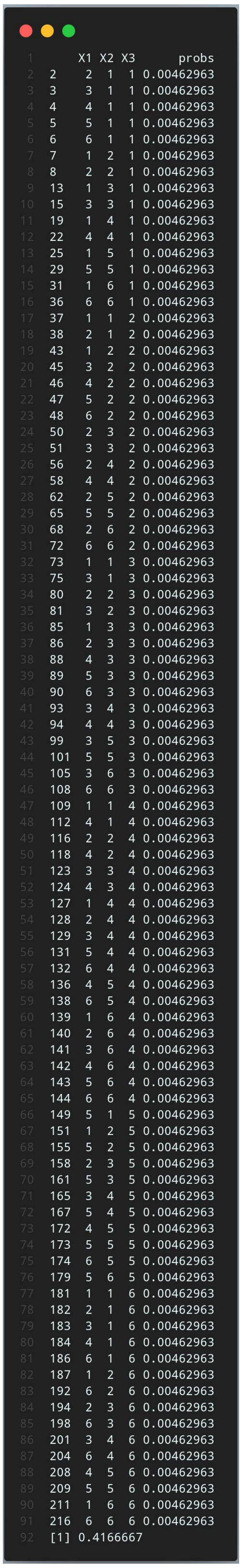
```
    X1 X2 X3      probs
2    2  1  1 0.00462963
3    3  1  1 0.00462963
4    4  1  1 0.00462963
5    5  1  1 0.00462963
6    6  1  1 0.00462963
7    1  2  1 0.00462963
8    2  2  1 0.00462963
13   1  3  1 0.00462963
15   3  3  1 0.00462963
19   1  4  1 0.00462963
22   4  4  1 0.00462963
25   1  5  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
137  5  5  4 0.00462963
142  4  6  4 0.00462963
144  6  6  4 0.00462963
145  1  1  5 0.00462963
149  5  1  5 0.00462963
152  2  2  5 0.00462963
155  5  2  5 0.00462963
159  3  3  5 0.00462963
161  5  3  5 0.00462963
166  4  4  5 0.00462963
167  5  4  5 0.00462963
169  1  5  5 0.00462963
170  2  5  5 0.00462963
171  3  5  5 0.00462963
172  4  5  5 0.00462963
174  6  5  5 0.00462963
179  5  6  5 0.00462963
180  6  6  5 0.00462963
181  1  1  6 0.00462963
186  6  1  6 0.00462963
188  2  2  6 0.00462963
192  6  2  6 0.00462963
195  3  3  6 0.00462963
198  6  3  6 0.00462963
202  4  4  6 0.00462963
204  6  4  6 0.00462963
209  5  5  6 0.00462963
210  6  5  6 0.00462963
211  1  6  6 0.00462963
212  2  6  6 0.00462963
213  3  6  6 0.00462963
214  4  6  6 0.00462963
215  5  6  6 0.00462963
[1] 0.4166667
```

d:output



```
    X1 X2 X3      probs
9    3  2  1 0.00462963
10   4  2  1 0.00462963
11   5  2  1 0.00462963
12   6  2  1 0.00462963
14   2  3  1 0.00462963
16   4  3  1 0.00462963
17   5  3  1 0.00462963
18   6  3  1 0.00462963
20   2  4  1 0.00462963
21   3  4  1 0.00462963
23   5  4  1 0.00462963
24   6  4  1 0.00462963
26   2  5  1 0.00462963
27   3  5  1 0.00462963
28   4  5  1 0.00462963
30   6  5  1 0.00462963
32   2  6  1 0.00462963
33   3  6  1 0.00462963
34   4  6  1 0.00462963
35   5  6  1 0.00462963
39   3  1  2 0.00462963
40   4  1  2 0.00462963
41   5  1  2 0.00462963
42   6  1  2 0.00462963
49   1  3  2 0.00462963
52   4  3  2 0.00462963
53   5  3  2 0.00462963
54   6  3  2 0.00462963
55   1  4  2 0.00462963
57   3  4  2 0.00462963
59   5  4  2 0.00462963
60   6  4  2 0.00462963
61   1  5  2 0.00462963
63   3  5  2 0.00462963
64   4  5  2 0.00462963
66   6  5  2 0.00462963
67   1  6  2 0.00462963
69   3  6  2 0.00462963
70   4  6  2 0.00462963
71   5  6  2 0.00462963
74   2  1  3 0.00462963
76   4  1  3 0.00462963
77   5  1  3 0.00462963
78   6  1  3 0.00462963
79   1  2  3 0.00462963
82   4  2  3 0.00462963
83   5  2  3 0.00462963
84   6  2  3 0.00462963
91   1  4  3 0.00462963
92   2  4  3 0.00462963
95   5  4  3 0.00462963
96   6  4  3 0.00462963
97   1  5  3 0.00462963
98   2  5  3 0.00462963
100  4  5  3 0.00462963
102  6  5  3 0.00462963
103  1  6  3 0.00462963
104  2  6  3 0.00462963
106  4  6  3 0.00462963
107  5  6  3 0.00462963
110  2  1  4 0.00462963
111  3  1  4 0.00462963
113  5  1  4 0.00462963
114  6  1  4 0.00462963
115  1  2  4 0.00462963
117  3  2  4 0.00462963
119  5  2  4 0.00462963
120  6  2  4 0.00462963
121  1  3  4 0.00462963
122  2  3  4 0.00462963
125  5  3  4 0.00462963
126  6  3  4 0.00462963
133  1  5  4 0.00462963
134  2  5  4 0.00462963
135  3  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
143  5  6  4 0.00462963
146  2  1  5 0.00462963
147  3  1  5 0.00462963
148  4  1  5 0.00462963
150  6  1  5 0.00462963
151  1  2  5 0.00462963
153  3  2  5 0.00462963
154  4  2  5 0.00462963
156  6  2  5 0.00462963
157  1  3  5 0.00462963
158  2  3  5 0.00462963
160  4  3  5 0.00462963
162  6  3  5 0.00462963
163  1  4  5 0.00462963
164  2  4  5 0.00462963
165  3  4  5 0.00462963
168  6  4  5 0.00462963
175  1  6  5 0.00462963
176  2  6  5 0.00462963
177  3  6  5 0.00462963
178  4  6  5 0.00462963
182  2  1  6 0.00462963
183  3  1  6 0.00462963
184  4  1  6 0.00462963
185  5  1  6 0.00462963
187  1  2  6 0.00462963
189  3  2  6 0.00462963
190  4  2  6 0.00462963
191  5  2  6 0.00462963
193  1  3  6 0.00462963
194  2  3  6 0.00462963
196  4  3  6 0.00462963
197  5  3  6 0.00462963
199  1  4  6 0.00462963
200  2  4  6 0.00462963
201  3  4  6 0.00462963
203  5  4  6 0.00462963
205  1  5  6 0.00462963
206  2  5  6 0.00462963
207  3  5  6 0.00462963
208  4  5  6 0.00462963
[1] 0.5555556
```

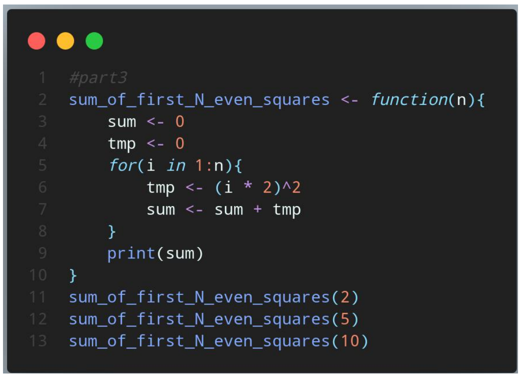
e:output



```
      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
```

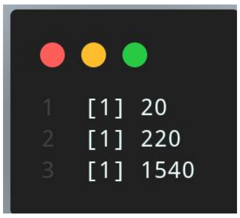
Part 3

code



```
#part3
sum_of_first_N_even_squares <- function(n){
    sum <- 0
    tmp <- 0
    for(i in 1:n){
        tmp <- (i * 2)^2
        sum <- sum + tmp
    }
    print(sum)
}
sum_of_first_N_even_squares(2)
sum_of_first_N_even_squares(5)
sum_of_first_N_even_squares(10)
```

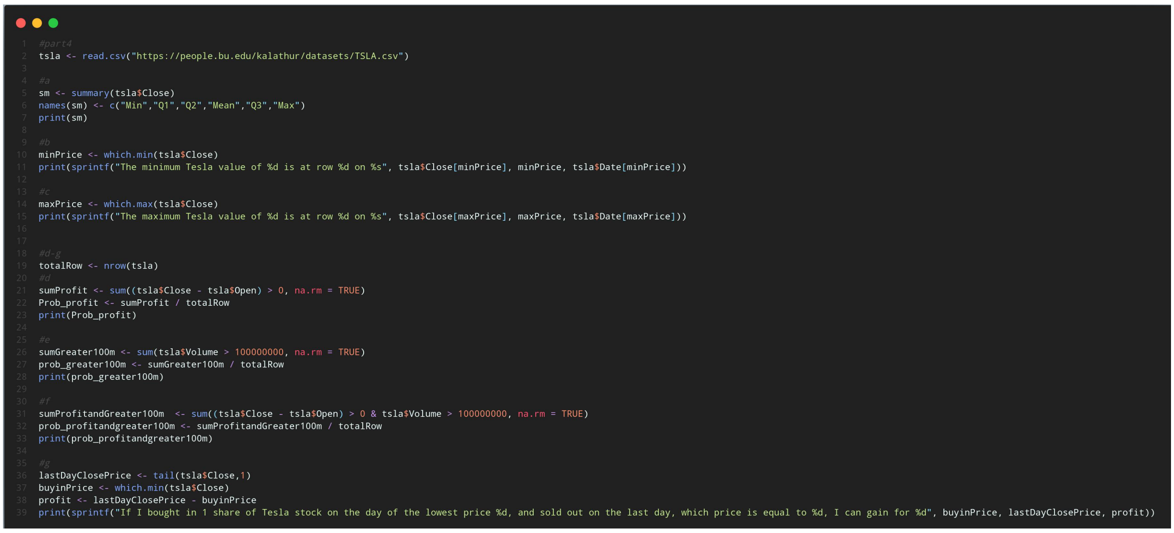
output



```
[1] 20
[1] 220
[1] 1540
```

Part 4

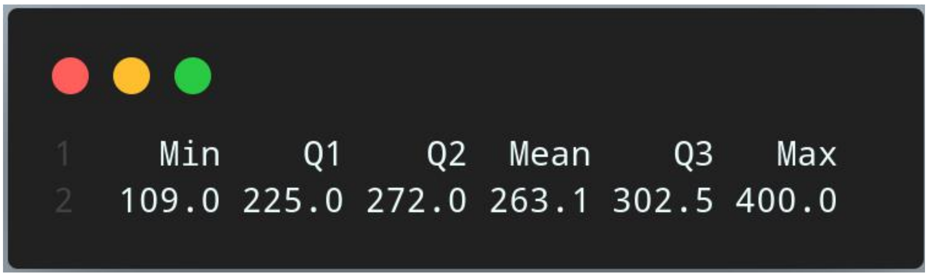
code



```
#part4
tsla <- read.csv("https://people.bu.edu/kalathur/datasets/TSLA.csv")

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

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

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


#d-g
totalRow <- nrow(tsla)
#d
sumProfit <- sum((tsla$Close - tsla$Open) > 0, na.rm = TRUE)
Prob_profit <- sumProfit / totalRow
print(Prob_profit)

#e
sumGreater100m <- sum(tsla$Volume > 100000000, na.rm = TRUE)
prob_greater100m <- sumGreater100m / totalRow
print(prob_greater100m)

#f
sumProfitandGreater100m  <- sum((tsla$Close - tsla$Open) > 0 & tsla$Volume > 100000000, na.rm = TRUE)
prob_profitandgreater100m <- sumProfitandGreater100m / totalRow
print(prob_profitandgreater100m)

#g
lastDayClosePrice <- tail(tsla$Close,1)
buyinPrice <- which.min(tsla$Close)
profit <- lastDayClosePrice - buyinPrice
print(sprintf("If I bought in 1 share of Tesla stock on the day of the lowest price %d, and sold out on the last day, which price is equal to %d, I can gain for %d", buyinPrice, lastDayClosePrice, profit))
```

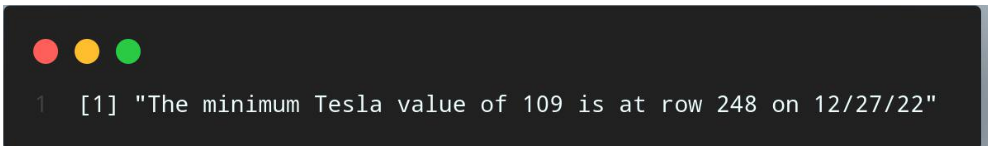
output

a:



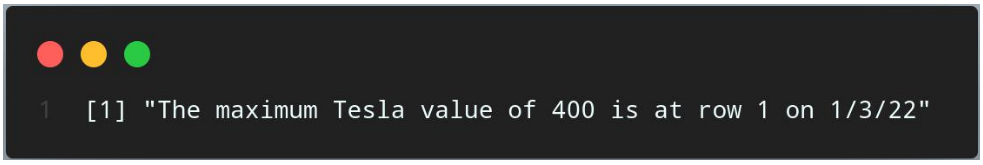
```
  Min    Q1    Q2  Mean    Q3   Max
109.0 225.0 272.0 263.1 302.5 400.0
```

b:



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

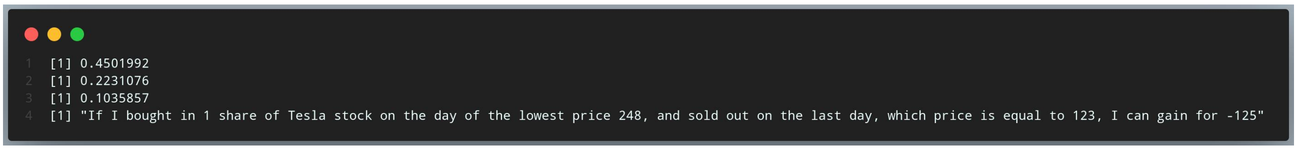
c:



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

d:e:f:g:

output:(d-g, from up to down)



```
[1] 0.4501992
[1] 0.2231076
[1] 0.1035857
[1] "If I bought in 1 share of Tesla stock on the day of the lowest price 248, and sold out on the last day, which price is equal to 123, I can gain for -125"
```