TUAN BUI

SBU ID: 113141951

AMS 380

HOMEWORK #1

<u>QUESTION 1:</u>

R code:

```
# Question 1a
# Generate a vector named 'myvec'
myvec <- c(1.5, 2.1, 1.8, 3.4, 2.6, 2.8, 0.9,1.9)
# A vector contains 1st, 3rd, 6nd elements in myvec
myvec[c(1,3,6)]

# Question 1b
# Print the elements in myvec which is greater than 2 and less than 3
myvec[myvec>2 & myvec<3]

# Question 1c
# Mean of myvec
mean(myvec)
# Sum of myvec
sum(myvec)

# Question 1d
# The amount of elements in myvec
length(myvec)
# The amount of elements in myvec less than 2.5
length(myvec[myvec<2.5])
```

Output from R:

```
> # Question 1a
> # Generate a vector named 'myvec'
> myvec <- c(1.5, 2.1, 1.8, 3.4, 2.6, 2.8, 0.9,1.9)

> # A vector contains 1st, 3rd, 6nd elements in myvec
> myvec[c(1,3,6)]
[1] 1.5 1.8 2.8

> # Question 1b
> # Print the elements in myvec which is greater than 2 and less than 3
> myvec[myvec>2 & myvec<3]
[1] 2.1 2.6 2.8

> # Question 1c
> # Mean of myvec
> mean(myvec)
[1] 2.125

> # Sum of myvec
> sum(myvec)
[1] 17

> # Question 1d
> # The amount of elements in myvec
> length(myvec)
[1] 8

> # The amount of elements in myvec less than 2.5
> length(myvec[myvec<2.5])
[1] 5
```

Answer:

1a. 1.5 1.8 2.8

1b. 2.1 2.6 2.8

1c. Mean: 2.125

Sum: 17

1d. There are 8 elements in myvec

There are 5 elements in myvec less than 2.5

QUESTION 2:

R code:

```
# Question 2


# load data set as dataframe

data_q2 <- read.table('~/OneDrive - Stony Brook University/SBU/MAT + AMS/Fall 2021/AMS
380/hw/01/HW1Q2.csv',header = T, sep = ',')


#Select the rows of data_q2 s.t. (a > 0.05 & b < 0.1)

subset(data_q2, a > 0.05 & b< 0.1)


#Select the rows of data_q2 s.t. (a > 0.05 or b < 0.1)

subset(data_q2, a > 0.05 | b< 0.1)


# Generate the now col 'c' in data_q2 s.t. c = b^2

transform(data_q2, c = b^2)
```

Output from R:

```
> # Question 2

> # load data set as dataframe
> data_q2 <- read.table('~/OneDrive - Stony Brook University/SBU/MAT + AMS/Fall 2021/AMS 380/hw/01/H ....
[TRUNCATED]

> #Select the rows of data_q2 s.t. (a > 0.05 & b < 0.1)
> subset(data_q2, a > 0.05 & b< 0.1)
           a          b
6  0.9769734  -0.8655129
8  1.0527115  -0.1971759
14 0.6365697 -0.3246859
16 0.5168620 -0.8953634
17 0.3689645 -1.3108015
21 2.1284519 -0.6111659
25 0.3104807 -0.2651451

> #Select the rows of data_q2 s.t. (a > 0.05 or b < 0.1)
> subset(data_q2, a > 0.05 | b< 0.1)
            a           b
2  -0.11945261 -0.45836533
3  -0.28039534 -1.06332613
4   0.56298953  1.26318518
5  -0.37243876 -0.34965039
6   0.97697339 -0.86551286
7  -0.37458086 -0.23627957
8   1.05271147 -0.19717589
10 -1.26015524  0.08473729
11  3.24103993  0.75405379
12 -0.41685759 -0.49929202
13  0.29822759  0.21444531
```

```
14  0.63656967 -0.32468591
15 -0.48378063  0.09458353
16  0.51686204 -0.89536336
17  0.36896453 -1.31080153
19  0.06529303  0.60070882
20 -0.03406725 -1.25127136
21  2.12845190 -0.61116592
22 -0.74133610 -1.18548008
25  0.31048075 -0.26514506

> # Generate the now col 'c' in data_q2 s.t. c = b^2
> transform(data_q2, c = b^2)
             a           b           c
1  -1.00837661  0.43652348 0.190552748
2  -0.11945261 -0.45836533 0.210098778
3  -0.28039534 -1.06332613 1.130662467
4   0.56298953  1.26318518 1.595636789
5  -0.37243876 -0.34965039 0.122255394
6   0.97697339 -0.86551286 0.749112515
7  -0.37458086 -0.23627957 0.055828035
8   1.05271147 -0.19717589 0.038878333
9  -1.04917701  1.10992029 1.231923050
10 -1.26015524  0.08473729 0.007180409
11  3.24103993  0.75405379 0.568597111
12 -0.41685759 -0.49929202 0.249292518
13  0.29822759  0.21444531 0.045986791
14  0.63656967 -0.32468591 0.105420941
15 -0.48378063  0.09458353 0.008946044
16  0.51686204 -0.89536336 0.801675543
17  0.36896453 -1.31080153 1.718200660
18 -0.21538051  1.99721338 3.988861304
19  0.06529303  0.60070882 0.360851091
```

20	-0.03406725	-1.25127136	1.565680020
21	2.12845190	-0.61116592	0.373523778
22	-0.74133610	-1.18548008	1.405363031
23	-1.09599627	2.19881035	4.834766950
24	0.03778840	1.31241298	1.722427821
25	0.31048075	-0.26514506	0.070301901

Answer:

2a.

	a	b
6	0.9769734	-0.8655129
8	1.0527115	-0.1971759
14	0.6365697	-0.3246859
16	0.5168620	-0.8953634
17	0.3689645	-1.3108015
21	2.1284519	-0.6111659
25	0.3104807	-0.2651451

2b.

	a	b
2	-0.11945261	-0.45836533
3	-0.28039534	-1.06332613
4	0.56298953	1.26318518
5	-0.37243876	-0.34965039
6	0.97697339	-0.86551286
7	-0.37458086	-0.23627957
8	1.05271147	-0.19717589
10	-1.26015524	0.08473729
11	3.24103993	0.75405379
12	-0.41685759	-0.49929202
13	0.29822759	0.21444531
14	0.63656967	-0.32468591
15	-0.48378063	0.09458353
16	0.51686204	-0.89536336
17	0.36896453	-1.31080153
19	0.06529303	0.60070882
20	-0.03406725	-1.25127136
21	2.12845190	-0.61116592
22	-0.74133610	-1.18548008
25	0.31048075	-0.26514506

2c.

	a	b	c
1	-1.00837661	0.43652348	0.190552748
2	-0.11945261	-0.45836533	0.210098778
3	-0.28039534	-1.06332613	1.130662467
4	0.56298953	1.26318518	1.595636789
5	-0.37243876	-0.34965039	0.122255394
6	0.97697339	-0.86551286	0.749112515
7	-0.37458086	-0.23627957	0.055828035
8	1.05271147	-0.19717589	0.038878333
9	-1.04917701	1.10992029	1.231923050
10	-1.26015524	0.08473729	0.007180409
11	3.24103993	0.75405379	0.568597111
12	-0.41685759	-0.49929202	0.249292518
13	0.29822759	0.21444531	0.045986791
14	0.63656967	-0.32468591	0.105420941
15	-0.48378063	0.09458353	0.008946044
16	0.51686204	-0.89536336	0.801675543
17	0.36896453	-1.31080153	1.718200660
18	-0.21538051	1.99721338	3.988861304
19	0.06529303	0.60070882	0.360851091
20	-0.03406725	-1.25127136	1.565680020
21	2.12845190	-0.61116592	0.373523778
22	-0.74133610	-1.18548008	1.405363031
23	-1.09599627	2.19881035	4.834766950
24	0.03778840	1.31241298	1.722427821
25	0.31048075	-0.26514506	0.070301901

<u>Question 3:</u>

a. (1) There are 3 doors, and the car is behind one of the doors; therefore, the probability that you pick the door having a car behind, and your strategy is to stay is: $\mathrm{P(winning)} = \frac{1}{3}$

(2) The 3 doors have the same probability to win which is $\frac{1}{3}$ each door, and they are independent variables. Therefore, the probability that the car behind the other doors that were not your first choice (i.e. the car is not behind the door you chose) is $\left(1 - \frac{1}{3}\right) = \frac{2}{3}$. Then, the host opens one of the two doors you didn't pick and which the car is not behind, so the probability that the car behind the door that is neither your choice nor opened by the host is $\frac{2}{3}$. Thus, if your strategy is to switch, your winning chance is: $\mathrm{P(winning)} = \frac{2}{3}$

Answer:

3a.1. If your strategy is to stay, your winning chance is: $\mathrm{P(winning)} = \frac{1}{3}$

3a.2. if your strategy is to switch, your winning chance is: $\mathrm{P(winning)} = \frac{2}{3}$

b. R code:

```r
n <- 0

m <- 0

door <- 1:3 #the door vector


#to stay

for (i in 1:1000) {

  selected_door <- sample(door,1) # the door you picked

  car_door <- sample(door,1) # the door which the car is behind


  if(selected_door == car_door){

    n <- n+1  } # count how many times that you pick the car door


  rm(selected_door,car_door)

}

print(n/1000) # the probability to win if you play 1000 times and your strategy is to stay


#to switch

for (j in 1:1000){

  selected_door <- sample(door,1) # the door which you pick at the beginning of the game

  car_door <- sample(door,1) # the door has the car behind

  repeat {

    opened_door <- sample(door,1) # the door that the host opens

    if(opened_door != selected_door && opened_door != car_door) break }

  new_door <- door[door != selected_door & door != opened_door] # the door that you switch your choice


  if(new_door == car_door){

    m <- m+1  } # count how many times that you pick the car door


  rm(selected_door,car_door,opened_door,new_door)

}

print (m/1000) # the probability to win if you play 1000 times and your strategy is to switch (i.e., always switch)
```

Output from R:

```
> n <- 0

> m <- 0

> door <- 1:3 #the door vector

> #to stay
> for (i in 1:1000) {
+   selected_door <- sample(door,1) # the door you picked
+   car_door <- sample(door,1) # the door which the car is  .... [TRUNCATED]

> print(n/1000) # the probability to win if you play 1000 times and your strategy is to stay
[1] 0.328

> #to switch
> for (j in 1:1000){
+   selected_door <- sample(door,1) # the door which you pick at the beginning of the game
+   car_door <- sample(do .... [TRUNCATED]

> print (m/1000) # the probability to win if you play 1000 times and your strategy is to switch (i.e. always switch)
[1] 0.67
```

Answer:

3b.1. 0.328

3b.2. 0.67