```
# A.
import pandas as pd

july4 = pd.read_csv("/content/july4_snapshot.csv")
july4
```

	visitor	day_pass	season_ticket	domestic	state	country	gender	age	maine_res	stay_four	payment_method	ice_cream_p
0	1	1	No	1	NY	USA	0	32	No	1	0	
1	2	0	Yes	1	Other	USA	1	43	No	1	0	
2	3	1	No	1	ME	USA	1	28	Yes	1	0	
3	4	0	Yes	1	NH	USA	1	35	No	0	0	
4	5	1	No	0	NaN	MEX	1	44	No	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	
5211	5212	0	Yes	1	NH	USA	0	37	No	1	0	
5212	5213	1	No	0	NaN	UK	1	30	No	1	0	
5213	5214	0	Yes	1	NH	USA	1	36	No	0	0	
5214	5215	0	Yes	1	ME	USA	0	38	Yes	1	0	
5215	5216	1	No	1	MA	USA	0	35	No	1	0	

5216 rows × 19 columns

```
#B:
july4.head()
```

	visitor	day_pass	season_ticket	domestic	state	country	gender	age	maine_res	stay_four	payment_method	ice_cream_purc
0	1	1	No	1	NY	USA	0	32	No	1	0	
1	2	0	Yes	1	Other	USA	1	43	No	1	0	
2	3	1	No	1	ME	USA	1	28	Yes	1	0	
3	4	0	Yes	1	NH	USA	1	35	No	0	0	
4	5	1	No	0	NaN	MEX	1	44	No	1	0	

5 now after the head() function

```
#CA
july4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5216 entries, 0 to 5215
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   visitor           5216 non-null   int64
 1   day_pass          5216 non-null   int64
 2   season_ticket     5216 non-null   object
 3   domestic          5216 non-null   int64
 4   state             4127 non-null   object
 5   country           5160 non-null   object
 6   gender            5216 non-null   int64
 7   age               5216 non-null   int64
 8   maine_res         5216 non-null   object
 9   stay_four         5216 non-null   int64
 10  payment_method    5216 non-null   int64
 11  ice_cream_purch   5216 non-null   int64
 12  ice_cream_flavor  5216 non-null   object
 13  sky_chair         5216 non-null   int64
 14  ferris_wheel      5216 non-null   int64
 15  lobster_claw      5216 non-null   int64
```

```
 16  lobster_junior    5216 non-null   int64
 17  merch_spend       5216 non-null   float64
 18  lobsterama_spend  5216 non-null   float64
dtypes: float64(2), int64(12), object(5)
memory usage: 774.4+ KB
```

D: Season ticket, state, country, Maine res, ice cream flavor ,gender,are categorical age, Merch spends, and lobstermen spend are numeric.

```
#e
july4.merch_spend.round(2)
```

```
0       34.53
1       23.81
2       49.23
3       55.51
4       61.02
        ...
5211    51.13
5212    43.17
5213    37.49
5214    45.34
5215    46.31
Name: merch_spend, Length: 5216, dtype: float64
```

```
#E
july4.isnull().sum().sum()/july4.size*100
```

```
1.1553519535033905
```

```
#F:
july4.isnull().sum()
```

```
visitor               0
day_pass              0
season_ticket         0
domestic              0
state              1089
country              56
gender                0
age                   0
maine_res             0
stay_four             0
payment_method        0
ice_cream_purch       0
ice_cream_flavor      0
sky_chair             0
ferris_wheel          0
lobster_claw          0
lobster_junior        0
merch_spend           0
lobsterama_spend      0
dtype: int64
```

```
# Fb
percent_missing = july4.isnull().sum() * 100 / len(july4)
percent_missing
```

```
visitor              0.000000
day_pass             0.000000
season_ticket        0.000000
domestic             0.000000
state               20.878067
country              1.073620
gender               0.000000
age                  0.000000
maine_res            0.000000
stay_four            0.000000
payment_method       0.000000
ice_cream_purch      0.000000
ice_cream_flavor     0.000000
sky_chair            0.000000
ferris_wheel         0.000000
lobster_claw         0.000000
lobster_junior       0.000000
merch_spend          0.000000
lobsterama_spend     0.000000
dtype: float64
```

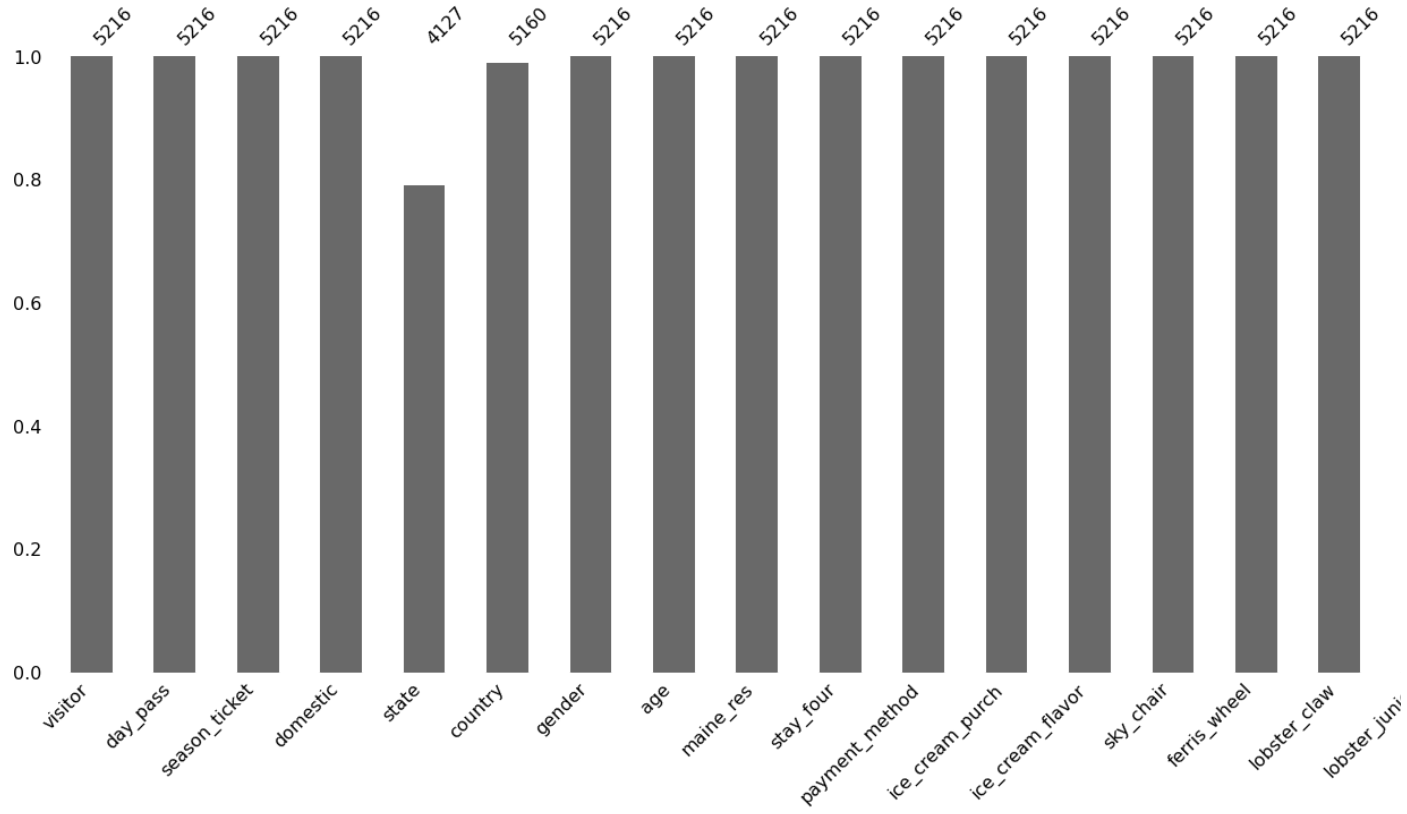
```
#Fa
total_percent_missing = 20.878067+1.073620
total_percent_missing
```

```
21.951687
```

```
#Fc
import missingno as msno#https://towardsdatascience.com/using-the-missingno-python-library-to-identify-and-visualise-missing-data-
msno.matrix(july4)
```

```
#Fd
msno.bar(july4)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f07be845790>
```




```
#fe
#state = july4[july4[state] == 'NaN']
state = july4[july4['state'].isna()]
state
```

	visitor	day_pass	season_ticket	domestic	state	country	gender	age	maine_res	stay_four	payment_method	ice_cream_p
4	5	1	No	0	NaN	MEX	1	44	No	1	0	
10	11	1	No	0	NaN	CAN	1	37	No	1	0	

NaN state data they all from other countries and have no season ticket.

14	15	1	No	0	NaN	PRA	1	40	No	1	0	

```
#Ga
#july4.filter(july4['age'] <= 15)
july4age= july4[(july4['age'] <= 15)]
#july4['age'] = july4['age'].clip(lower=15)
july4age
```

	visitor	day_pass	season_ticket	domestic	state	country	gender	age	maine_res	stay_four	payment_method	ice_cream_p
1352	1353	0	Yes	1	Other	USA	1	14	No	1	1	

```
july4.loc[july4['age'] <= 15, 'age'] = 15
july4.loc[july4['age'] <= 15, 'age']
```

```
1352    15
Name: age, dtype: int64
```

```
july4.loc[ july4['age'] == 15, 'age'] = 15
```

```
#Ha
july4hour= july4[(july4['stay_four'] >= 1)]
july4hour
```

	visitor	day_pass	season_ticket	domestic	state	country	gender	age	maine_res	stay_four	payment_method	ice_cream_p
0	1	1	No	1	NY	USA	0	32	No	1	0	
1	2	0	Yes	1	Other	USA	1	43	No	1	0	
2	3	1	No	1	ME	USA	1	28	Yes	1	0	
4	5	1	No	0	NaN	MEX	1	44	No	1	0	
7	8	0	Yes	1	VT	USA	0	29	No	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...
5207	5208	1	No	0	NaN	CAN	1	35	No	1	0	
5211	5212	0	Yes	1	NH	USA	0	37	No	1	0	
5212	5213	1	No	0	NaN	UK	1	30	No	1	0	
5214	5215	0	Yes	1	ME	USA	0	38	Yes	1	0	
5215	5216	1	No	1	MA	USA	0	35	No	1	0	

3126 rows × 19 columns

The percentage of guests from the entire dataset who stayed at Lobster Land for more than four hours on July 4th is 59.93%

```
len(july4hour['stay_four'])/len(july4['stay_four'])*100
```

```
59.93098159509203
```

```
#hb
df = pd.DataFrame(july4hour)
```

```
domestic_visitors= df.loc[df['country'] =='USA']
domestic_4 = july4.loc[july4['country'] =='USA']
international_visitors= df.loc[df['country'] !='USA']
international_4 = july4.loc[july4['country'] !='USA']
```

The ratio of domestic visitors who stayed for more than 4 hours on that day is 52.047492125030296% The ratio of international visitors who stayed for more than 4 hours on that day is 89.80716253443526

```
domestic_visitors_ratio = len(domestic_visitors['country'])/len(domestic_4['country'])*100
domestic_visitors_ratio
```

```
52.047492125030296
```

```
international_visitors_ratio = len(international_visitors['country'])/len(international_4['country'])*100
international_visitors_ratio# think about : why this number is different ,.
```

```
89.80716253443526
```

HC: For my stepB , mine is different. There are several reasons. 1: there are some missing data when people collect. 2: most visitors are from the USA because the park is in the USA. Then some visitors want to protect their privacy , they tell then people the wrong information

```
# IA
july4.columns
remove_maineres = july4.drop("maine_res", axis = 1)
remove_maineres#https://sparkbyexamples.com/pandas/pandas-delete-rows-based-on-column-value/#:~:text=Use%20drop()%20method%20to,or
```

	visitor	day_pass	season_ticket	domestic	state	country	gender	age	stay_four	payment_method	ice_cream_purch	ice_c
0	1	1	No	1	NY	USA	0	32	1	0	1	
1	2	0	Yes	1	Other	USA	1	43	1	0	1	
2	3	1	No	1	ME	USA	1	28	1	0	0	
3	4	0	Yes	1	NH	USA	1	35	0	0	0	
4	5	1	No	0	NaN	MEX	1	44	1	0	1	
...	...	...	...	...	...	...	...	...	...	...	...	...
5211	5212	0	Yes	1	NH	USA	0	37	1	0	0	
5212	5213	1	No	0	NaN	UK	1	30	1	0	1	
5213	5214	0	Yes	1	NH	USA	1	36	0	0	1	
5214	5215	0	Yes	1	ME	USA	0	38	1	0	0	
5215	5216	1	No	1	MA	USA	0	35	1	0	0	

5216 rows × 18 columns

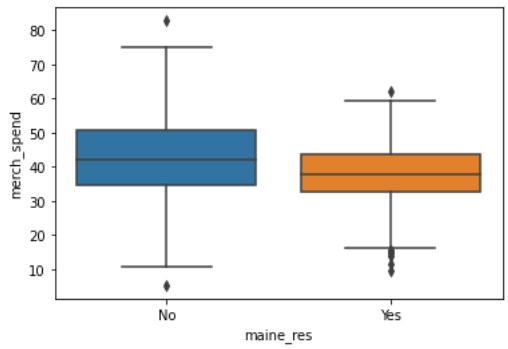
I have 2 reasons, first of all, this park affects plenty of visitors, and they have data called “county” and we can use groupby or filter to get the Maine res data.

```
#Ja
# https://www.geeksforgeeks.org/python-pandas-dataframe-rename/
renamedf = july4.rename(columns={'stay_four': 'stay_4'})
renamedf
```

	visitor	day_pass	season_ticket	domestic	state	country	gender	age	maine_res	stay_4	payment_method	ice_cream_purc
0	1	1	No	1	NY	USA	0	32	No	1	0	
1	2	0	Yes	1	Other	USA	1	43	No	1	0	
2	3	1	No	1	ME	USA	1	28	Yes	1	0	
3	4	0	Yes	1	NH	USA	1	35	No	0	0	
4	5	1	No	0	NaN	MEX	1	44	No	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	
5211	5212	0	Yes	1	NH	USA	0	37	No	1	0	

```
#K
import pandas as pd
import seaborn as sns
#https://seaborn.pydata.org/generated/seaborn.boxplot.html
sns.boxplot(data=july4, x="maine_res", y="merch_spend")# ana: median and 3rd is higher than other
# why: place , maybe not maybe people from maine is easy to back home and they need not mache in the losbster land
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f07c1133670>
```



Ka:From the boxplot, it is shown that plenty of people say no and the median of the boxplot is more than the people who say yes. ana: meadian and 3rd is higher than other why: place , maybe not maybe people from maine is easy to back home and they need not mache in the losbster land

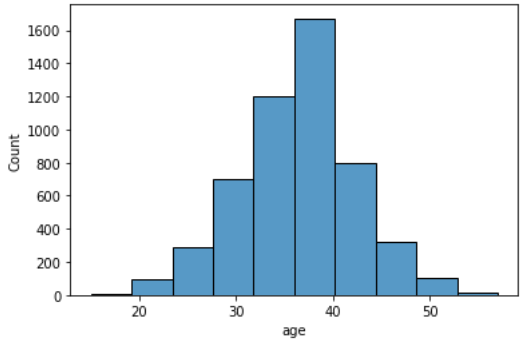
```
#L

#sky_chair  ferris_wheel    lobster_claw    lobster_junior
#sum function
#use maybe seaborn show indicate of each of those
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
sum_of_4 = {'Sky Chairs': july4['sky_chair'].sum(),
         'Ferris Wheel': july4['ferris_wheel'].sum(),
         'Lobster Claw': july4['lobster_claw'].sum(),
         'Lobster Junior': july4['lobster_junior'].sum()}
df = pd.DataFrame(sum_of_4, index=[0])
sns.barplot(data=df)
plt.ylabel('Total games')
plt.title('Total game Type')
```

the most popular projram is skychair, and the less popular one is Lobster Claw

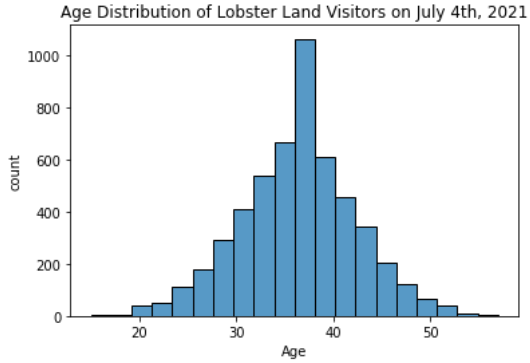
MA:use the histplot function and specify the number of bins using the bins parameter. change the bin=?

```
#Ma
sns.histplot(data=july4['age'], bins=10)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f07be4f3bb0>
```

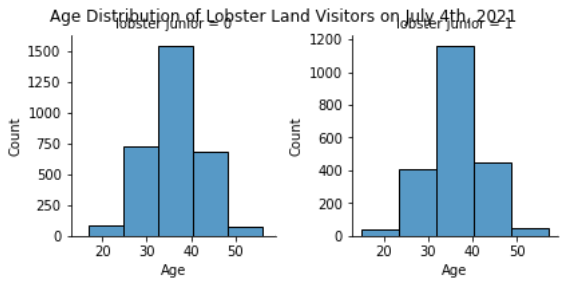



```
#Mb
sns.histplot(data=july4['age'], bins=20)
plt.xlabel('Age')
plt.ylabel('count')
plt.title('Age Distribution of Lobster Land Visitors on July 4th, 2021')
plt.show()
```



Mc: the difference between a and b chart, there are more bars in second chart and show more detail of each bar. the y axis reduce the number.

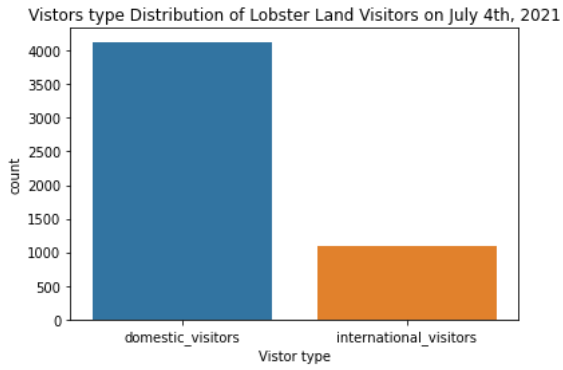
```
#lobster_junior = july4['lobster_junior']
#age=july4['age']

df = pd.DataFrame({
    'Age': july4['age'],
    'lobster junior': july4['lobster_junior']})
g = sns.FacetGrid(df, col='lobster junior', sharey=False)
g.map(sns.histplot, 'Age', bins=5)
g.fig.suptitle('Age Distribution of Lobster Land Visitors on July 4th, 2021')
plt.show()
#https://seaborn.pydata.org/generated/seaborn.FacetGrid.html
```



Mc: yes , becuse this activity attract people not like the big roller. So people between 25 to 45 like it. it is not too dangerous.

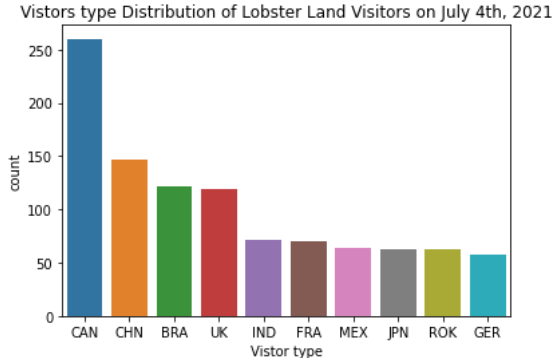
```
#N
df = pd.DataFrame(july4)
domestic_visitors= df.loc[df['country'] =='USA']
international_visitors= df.loc[df['country'] !='USA']

in_do = {'domestic_visitors': len(domestic_visitors['visitor']),
         'international_visitors': len(international_visitors['visitor'])}
df = pd.DataFrame(in_do, index=[0])
sns.barplot(data=df)
plt.ylabel('count')
plt.xlabel('Vistor type')
plt.title('Vistors type Distribution of Lobster Land Visitors on July 4th, 2021')
```

```
Text(0.5, 1.0, 'Vistors type Distribution of Lobster Land Visitors on July 4th, 2021')
```




```
df = pd.DataFrame(july4)
international_visitors= df.loc[df['country'] !='USA']

# sort by count in descending order
#https://www.statology.org/frequency-tables-python/
#sns.countplot(data=international_visitors, x='country')
sns.countplot(data=international_visitors, x='country', order=international_visitors['country'].value_counts().index)
plt.ylabel('count')
plt.xlabel('Vistor type')
plt.title('Vistors type Distribution of Lobster Land Visitors on July 4th, 2021')
# filter to just remove the USA,
```

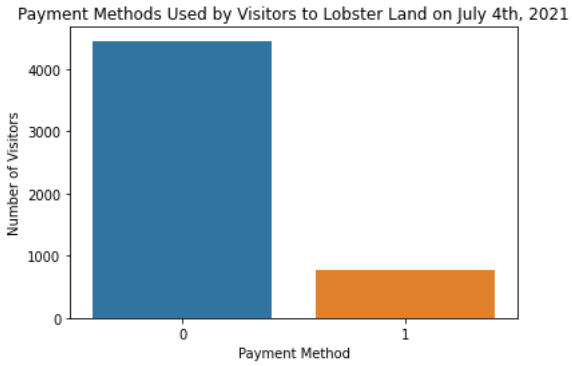
```
Text(0.5, 1.0, 'Vistors type Distribution of Lobster Land Visitors on July 4th,
2021')
```



Plenty of people are domestic vistors, because people in USA are more easy to go to the park, and they have advantage on location, when I least all it is hard to see , so I make another plot。 Canada is the most vistor of all the international vistor

```
#Ob
#
df3 = pd.DataFrame(july4)
my_tab = pd.crosstab(index=df['payment_method'],  # Make a crosstab
                     columns="count")                  # Name the count column


df3 = pd.DataFrame(my_tab)
# sort by count in descending order
df3 = df3.sort_values(by='count', ascending=False)
```

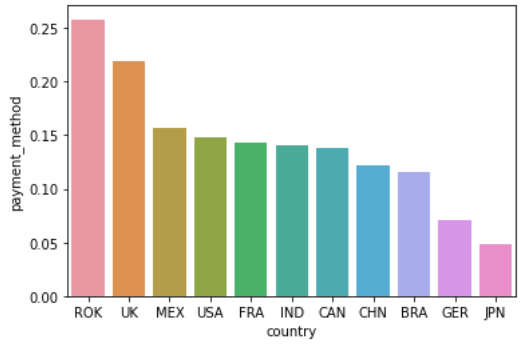
```
#https://www.statology.org/frequency-tables-python/
# create barplot using seaborn
sns.barplot(data=df3, x=df3.index, y='count')
plt.xlabel('Payment Method')
plt.ylabel('Number of Visitors')
plt.title('Payment Methods Used by Visitors to Lobster Land on July 4th, 2021')
plt.show()
#sns.countplot(data=df3, x='payment_method')

df3
```



payment_method	col_0 count
0	4454
1	762

```
df = pd.DataFrame(july4)
cash_pm= df.loc[df['payment_method'] ==1]
sub_data = df[['country', 'payment_method']]
cash_payments = sub_data[sub_data['payment_method'] ==1].groupby('country').count()
prop_cash_payments = cash_payments / sub_data.groupby('country').count()
# Create the bar plot
#sns.barplot(x=prop_cash_payments.index, y='payment_method', data=prop_cash_payments.reset_index())

sns.barplot(x='country', y='payment_method', data=july4, order=july4.groupby('country')['payment_method'].mean().sort_values(ascer
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f07bc4e20d0>
```



Ob: it is show that most people from ROK, UK Mex use cash buy ticket the ratio of the payment of the most people is ROK, UK Mex , and they buy ticket by cash

---

Part 3: For the past three days, I decided to track the number of Wechat steps I take each day using my phone's pedometer. I chose this metric because I have been trying to be more active and I wanted to see if I was meeting my daily activity goal. Over the three days, I found that my step count varied significantly from day to day, ranging from 7177， 7423 to 10148 steps. I noticed that on the day where I walked the least, I spent more time sitting at my desk, and that day was also the day where I felt the least productive. On the days where I hit or exceeded my step goal, I noticed that I had more energy and felt better overall. I went to 2 office hour and 2 class on that day. This tracking has made me more aware of my daily activity levels, and I plan to continue tracking my step count in the future. It also encouraged me to take breaks from sitting and incorporate more movement into my daily routine. While nobody around me reacted to what I was doing, I found this experiment to be quite insightful and I plan on continuing to track my steps to maintain a healthy and active lifestyle. (from wechat)

✓ 2s completed at 3:22 PM