In [1]:
```
#Author: Vinay Maruri

#Problem Set 2: Economics 121
#Instead of writing this problem set out by hand, I have chosen to complete th
e problem set in Python. Questions 5 and 6 were answered using TA Chris Campo
s's STATA code provided in section.
```

In [2]:
```
import seaborn as sns
import csv
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import zipfile
from pathlib import Path


# Default plot configurations
%matplotlib inline
plt.rcParams['figure.figsize'] = (16,8)
plt.rcParams['figure.dpi'] = 150
sns.set()

from IPython.display import display, Latex, Markdown
```

In [3]:
```
excel_file = 'Anne.xls'
chocolates = pd.read_excel(excel_file)
```

In [4]:
```
chocolates
```

Out[4]:

	price ($)	dayofweek	timestamp	mobile (1=mobile, 0=not)	Safari (1=yes, 0=not)	Buy (1=buy, 0=not buy)
0	1.3	Mon	17:42	1	0	1
1	1.9	Tue	13:36	0	1	0
2	0.5	Mon	09:31	0	0	1
3	1.1	Fri	14:22	1	0	1
4	0.7	Tue	11:56	0	0	1
5	0.6	Wed	06:19	0	0	1
6	0.8	Mon	13:44	1	0	1
7	1.1	Wed	23:12	0	0	0
8	1.2	Thu	22:16	0	0	0
9	1.5	Thu	17:28	0	0	0
10	1.3	Thu	12:22	0	0	0
11	1.8	Mon	10:37	0	0	0
12	0.9	Sat	23:33	0	0	1
13	2.2	Mon	17:17	0	0	0
14	0.5	Thu	10:21	1	0	1
15	1.8	Tue	10:59	0	0	0
16	1.3	Wed	23:35	0	0	1
17	1.6	Fri	10:46	0	1	0
18	0.7	Thu	08:08	0	0	1
19	0.8	Tue	09:40	1	0	1
20	2.1	Thu	21:19	0	0	0
21	2.4	Wed	20:10	1	1	1
22	1.1	Sat	14:45	0	1	1
23	1.8	Tue	21:52	0	1	1
24	2.2	Sat	16:21	0	0	0
25	2.2	Thu	14:16	0	1	0
26	0.6	Thu	08:20	0	0	1
27	0.5	Tue	22:50	0	1	1
28	0.8	Tue	11:30	0	0	1
29	2.2	Thu	18:19	0	0	0
...	...	...	...	...	...	...
993	1.1	Sat	16:03	1	0	1
994	1.6	Tue	22:07	0	0	0
995	2.4	Sat	19:32	1	0	0
996	1.1	Wed	12:25	0	0	1

	price ($)	dayofweek	timestamp	mobile (1=mobile, 0=not)	Safari (1=yes, 0=not)	Buy (1=buy, 0=not buy)
997	1.8	Thu	20:12	0	0	0
998	1.1	Sat	23:20	0	0	1
999	2.0	Tue	22:56	1	0	1
1000	1.1	Sun	20:50	0	0	1
1001	2.2	Sat	11:19	1	0	0
1002	2.0	Fri	13:26	0	0	0
1003	2.0	Tue	17:10	1	1	1
1004	1.4	Fri	13:37	0	0	0
1005	0.9	Tue	15:06	0	0	1
1006	0.5	Sat	21:52	0	0	1
1007	1.1	Mon	08:04	1	0	1
1008	0.5	Tue	23:43	0	0	1
1009	1.8	Thu	20:05	1	0	1
1010	1.8	Sat	22:38	0	0	0
1011	1.3	Thu	21:38	0	0	1
1012	2.0	Sat	14:38	0	0	0
1013	1.5	Sun	23:25	1	0	1
1014	0.5	Thu	11:03	1	0	1
1015	0.7	Sat	06:55	1	0	1
1016	0.8	Thu	07:56	1	0	1
1017	1.1	Fri	10:00	1	0	1
1018	0.6	Tue	16:46	1	0	1
1019	1.7	Mon	10:28	0	0	0
1020	1.5	Sun	22:21	0	0	0
1021	1.7	Sat	06:11	0	0	0
1022	1.3	Wed	20:55	0	0	1

1023 rows × 6 columns

In [5]:
```
chocolates.head()
```

Out[5]:

	price ($)	dayofweek	timestamp	mobile (1=mobile, 0=not)	Safari (1=yes, 0=not)	Buy (1=buy, 0=not buy)
0	1.3	Mon	17:42	1	0	1
1	1.9	Tue	13:36	0	1	0
2	0.5	Mon	09:31	0	0	1
3	1.1	Fri	14:22	1	0	1
4	0.7	Tue	11:56	0	0	1

In [6]:
```
chocolates['dayofweek'] = chocolates['dayofweek'].map({'Mon': 0, 'Tue': 1, 'Wed': 2, 'Thu': 3, 'Fri': 4, 'Sat': 5, 'Sun': 6})
```

In [7]:
```
chocolates.groupby('dayofweek').head()
```

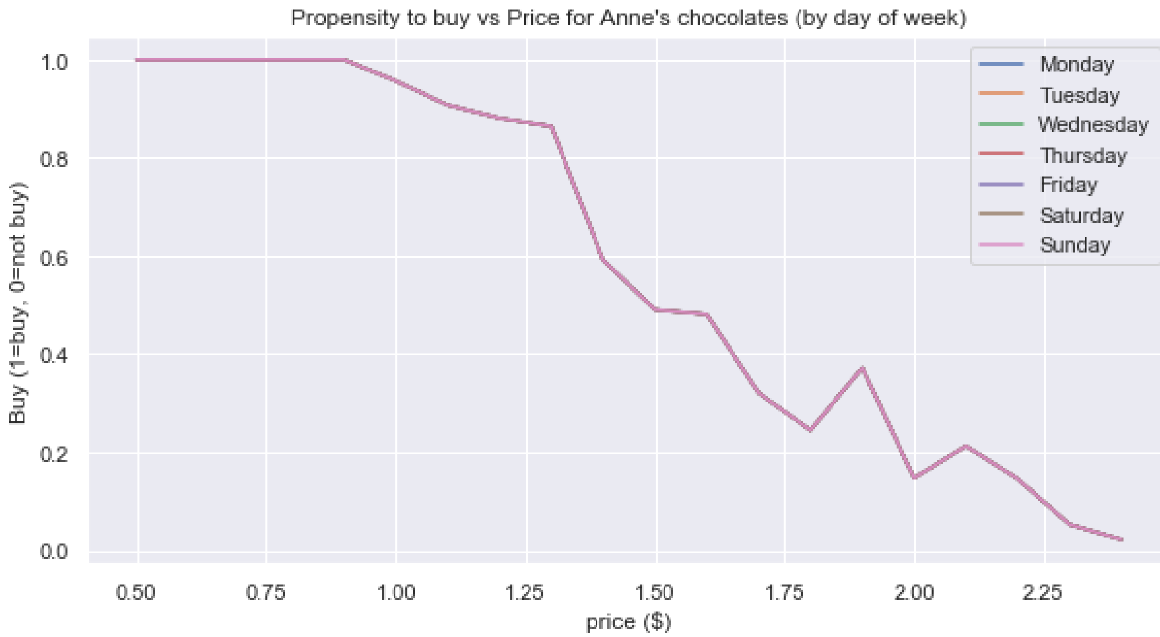
Out[7]:

	price ($)	dayofweek	timestamp	mobile (1=mobile, 0=not)	Safari (1=yes, 0=not)	Buy (1=buy, 0=not buy)
0	1.3	0	17:42	1	0	1
1	1.9	1	13:36	0	1	0
2	0.5	0	09:31	0	0	1
3	1.1	4	14:22	1	0	1
4	0.7	1	11:56	0	0	1
5	0.6	2	06:19	0	0	1
6	0.8	0	13:44	1	0	1
7	1.1	2	23:12	0	0	0
8	1.2	3	22:16	0	0	0
9	1.5	3	17:28	0	0	0
10	1.3	3	12:22	0	0	0
11	1.8	0	10:37	0	0	0
12	0.9	5	23:33	0	0	1
13	2.2	0	17:17	0	0	0
14	0.5	3	10:21	1	0	1
15	1.8	1	10:59	0	0	0
16	1.3	2	23:35	0	0	1
17	1.6	4	10:46	0	1	0
18	0.7	3	08:08	0	0	1
19	0.8	1	09:40	1	0	1
21	2.4	2	20:10	1	1	1
22	1.1	5	14:45	0	1	1
23	1.8	1	21:52	0	1	1
24	2.2	5	16:21	0	0	0
30	0.6	6	23:50	0	0	1
32	2.4	2	23:39	0	1	0
33	1.5	5	19:03	0	0	1
34	1.8	4	08:34	1	0	1
38	0.5	4	19:46	1	0	1
39	2.0	5	06:31	0	0	0
41	1.9	4	12:22	0	0	0
43	2.0	6	22:05	0	0	0
63	1.3	6	13:43	1	1	1
68	2.3	6	19:29	0	1	0
120	0.5	6	15:19	1	0	1

1) Draw the monthly demand curve for Anne' chocolates.

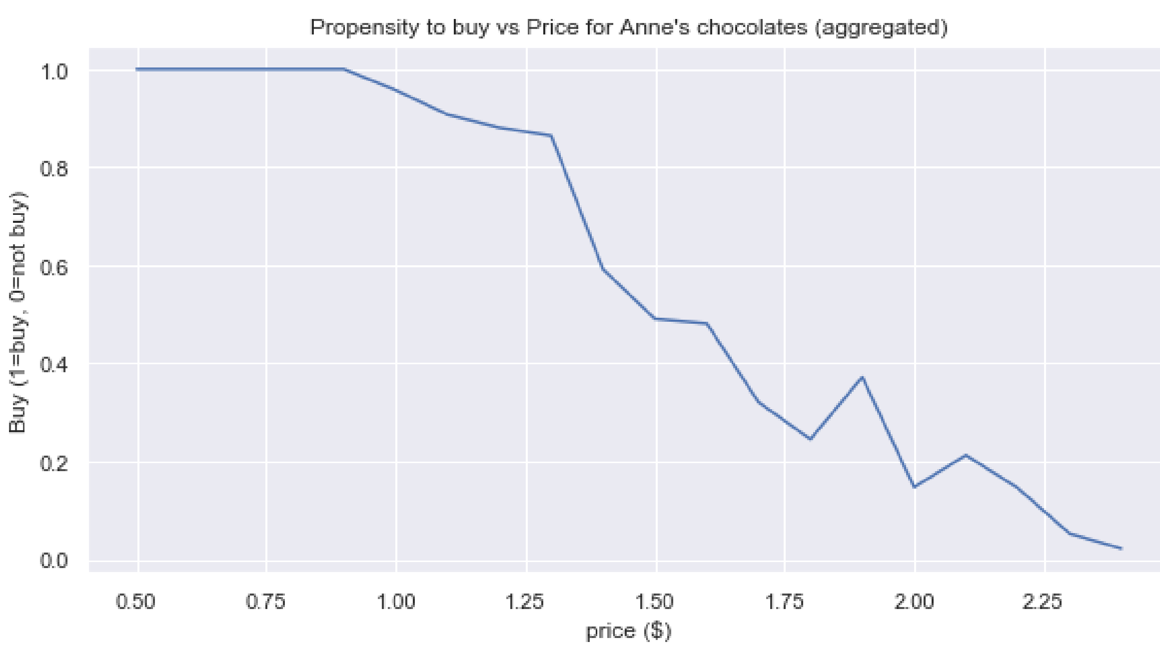
In [8]:
```
plt.figure(figsize = (10,5))

newdf0 = chocolates.loc[chocolates['dayofweek'] == 0]
ax = sns.lineplot(x = chocolates['price ($)'], y = chocolates['Buy (1=buy, 0=n
ot buy)'], data = newdf0, ci = None)

newdf1 = chocolates.loc[chocolates['dayofweek'] == 1]
ay = sns.lineplot(x = chocolates['price ($)'], y = chocolates['Buy (1=buy, 0=n
ot buy)'], data = newdf1, ci = None)

newdf2 = chocolates.loc[chocolates['dayofweek'] == 2]
az = sns.lineplot(x = chocolates['price ($)'], y = chocolates['Buy (1=buy, 0=n
ot buy)'], data = newdf2, ci = None)

newdf3 = chocolates.loc[chocolates['dayofweek'] == 3]
aa = sns.lineplot(x = chocolates['price ($)'], y = chocolates['Buy (1=buy, 0=n
ot buy)'], data = newdf3, ci = None)

newdf4 = chocolates.loc[chocolates['dayofweek'] == 4]
ab = sns.lineplot(x = chocolates['price ($)'], y = chocolates['Buy (1=buy, 0=n
ot buy)'], data = newdf4, ci = None)

newdf5 = chocolates.loc[chocolates['dayofweek'] == 5]
ac = sns.lineplot(x = chocolates['price ($)'], y = chocolates['Buy (1=buy, 0=n
ot buy)'], data = newdf5, ci = None)

newdf6 = chocolates.loc[chocolates['dayofweek'] == 6]
ad = sns.lineplot(x = chocolates['price ($)'], y = chocolates['Buy (1=buy, 0=n
ot buy)'], data = newdf6, ci = None)

plt.title("Propensity to buy vs Price for Anne's chocolates (by day of week)")
plt.legend(labels = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
'Saturday', 'Sunday'])


#This is a plot of monthly demand curves for Anne's Chocolates, separated by d
ay of week.
```

Out[8]: <matplotlib.legend.Legend at 0x28900dad9b0>



In [9]:
```
plt.figure(figsize = (10,5))

ax = sns.lineplot(x = chocolates['price ($)'], y = chocolates['Buy (1=buy, 0=n
ot buy)'], data = chocolates, ci = None)

plt.title("Propensity to buy vs Price for Anne's chocolates (aggregated)")

#This is a plot of Propensity to buy vs Price for Anne's chocolates, with all
 price and buy data aggregated, and not separated by day of week.
```

Out[9]: Text(0.5,1,"Propensity to buy vs Price for Anne's chocolates (aggregated)")



Remark: There appears to be no difference between day of week with regards to the propensity to buy Anne's Chocolates. There also appears to be no difference between the aggregated curve and the curves separated by day of week.

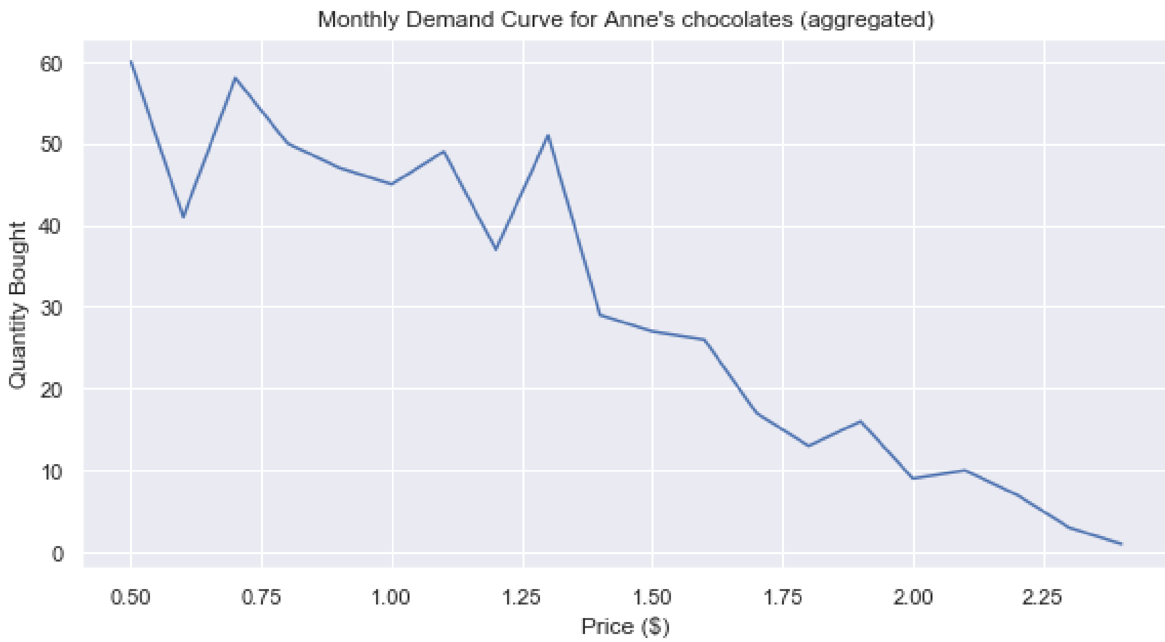
In [10]:
```
aggchoc = chocolates.groupby('price ($)')
aggchoc = aggchoc.agg({'Buy (1=buy, 0=not buy)': 'sum'})
aggchoc['price ($)'] = aggchoc.index
aggchoc
#Note: The buy column here is actually quantity bought- I aggregated the 1's a
nd 0's to form the total number of Anne's chocolates bought at a given price.
```

Out[10]:

	Buy (1=buy, 0=not buy)	price ($)
price ($)		
0.5	60	0.5
0.6	41	0.6
0.7	58	0.7
0.8	50	0.8
0.9	47	0.9
1.0	45	1.0
1.1	49	1.1
1.2	37	1.2
1.3	51	1.3
1.4	29	1.4
1.5	27	1.5
1.6	26	1.6
1.7	17	1.7
1.8	13	1.8
1.9	16	1.9
2.0	9	2.0
2.1	10	2.1
2.2	7	2.2
2.3	3	2.3
2.4	1	2.4

In [11]:
```
plt.figure(figsize = (10,5))

ax = sns.lineplot(x = aggchoc['price ($)'], y = aggchoc['Buy (1=buy, 0=not bu
y)'], data = aggchoc, ci = None)

plt.title("Monthly Demand Curve for Anne's chocolates (aggregated)")

plt.ylabel('Quantity Bought')
plt.xlabel('Price ($)')

#This is a plot of Monthly demand curve for Anne's chocolates, with all price
 and buy data aggregated, and not separated by day of week.
```

Out[11]:
```
Text(0.5,0,'Price ($)')
```



2) What would be the uniform price that you would advise Anne to charge?

In [12]:
```
aggchoc['revenue'] = aggchoc['Buy (1=buy, 0=not buy)'] * aggchoc['price ($)']
aggchoc['cost'] = aggchoc['Buy (1=buy, 0=not buy)'] * 1
aggchoc['profit'] = aggchoc['revenue'] - aggchoc['cost']
aggchoc
#Note: It is specified in the problem that it costs Anne $1 to make chocolate.
 I am interpreting that as marginal cost.
```

Out[12]:

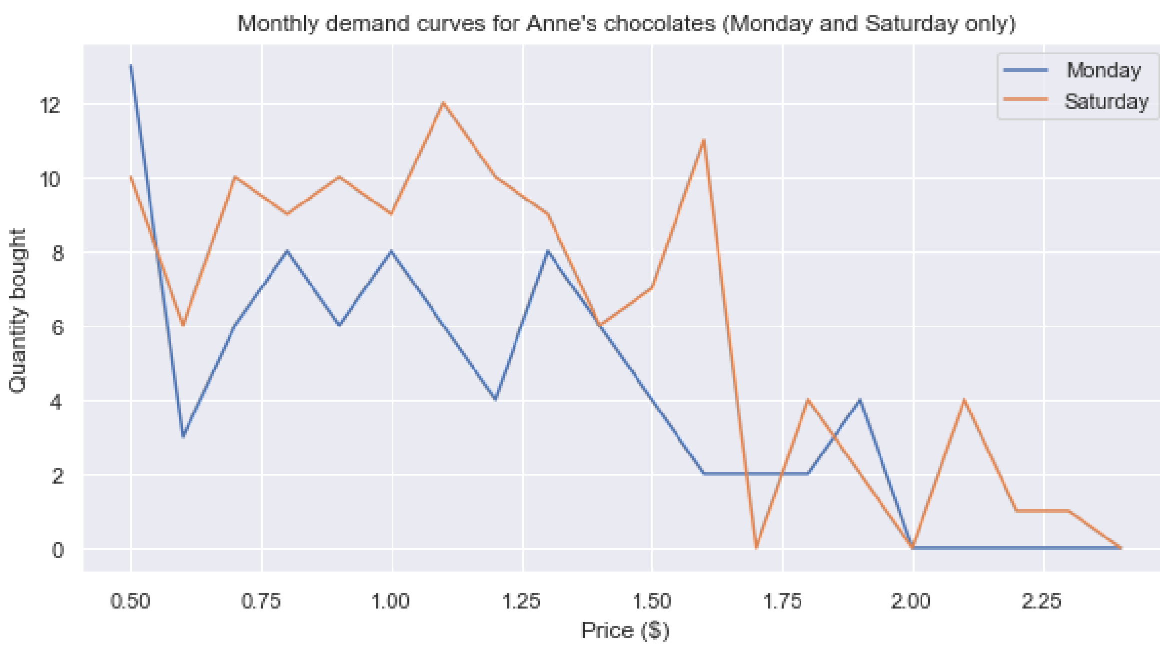
price ($)	Buy (1=buy, 0=not buy)	price ($)	revenue	cost	profit
0.5	60	0.5	30.000000	60	-30.000000
0.6	41	0.6	24.600001	41	-16.399999
0.7	58	0.7	40.599999	58	-17.400001
0.8	50	0.8	40.000001	50	-9.999999
0.9	47	0.9	42.299999	47	-4.700001
1.0	45	1.0	45.000000	45	0.000000
1.1	49	1.1	53.900001	49	4.900001
1.2	37	1.2	44.400002	37	7.400002
1.3	51	1.3	66.299998	51	15.299998
1.4	29	1.4	40.599999	29	11.599999
1.5	27	1.5	40.500000	27	13.500000
1.6	26	1.6	41.600001	26	15.600001
1.7	17	1.7	28.900001	17	11.900001
1.8	13	1.8	23.399999	13	10.399999
1.9	16	1.9	30.400000	16	14.400000
2.0	9	2.0	18.000000	9	9.000000
2.1	10	2.1	20.999999	10	10.999999
2.2	7	2.2	15.400000	7	8.400000
2.3	3	2.3	6.900000	3	3.900000
2.4	1	2.4	2.400000	1	1.400000

It appears that Anne's profit is maximized when price is set at $1.30. Hence, this is the uniform price I would advise Anne to charge.

3) Draw the demand curve that corresponds to customers who visit her site on Mondays. Draw the demand curve that corresponds to customers who visit her site on Saturdays.

In [13]:
```
plt.figure(figsize = (10, 5))

disaggchocmon = chocolates.loc[chocolates['dayofweek'] == 0]
disaggchocsat = chocolates.loc[chocolates['dayofweek'] == 5]

disaggchocsat = disaggchocsat.groupby('price ($)').agg({'Buy (1=buy, 0=not bu
y)': 'sum'})
disaggchocmon = disaggchocmon.groupby('price ($)').agg({'Buy (1=buy, 0=not bu
y)': 'sum'})

disaggchocmon['price'] = disaggchocmon.index
disaggchocsat['price'] = disaggchocsat.index

ax = sns.lineplot(x='price', y='Buy (1=buy, 0=not buy)', data = disaggchocmon)
ay = sns.lineplot(x='price', y='Buy (1=buy, 0=not buy)', data = disaggchocsat)

plt.title("Monthly demand curves for Anne's chocolates (Monday and Saturday on
ly)")
plt.legend(labels = ['Monday', 'Saturday'])
plt.ylabel('Quantity bought')
plt.xlabel('Price ($)')
```

Out[13]:
```
Text(0.5,0,'Price ($)')
```



4) Suppose that you decide to advise Anne to set different prices on different day of the week. What price would you recommend her to set on different days of the week?

In [14]:
```
disaggchoctue = chocolates.loc[chocolates['dayofweek'] == 1]
disaggchocwed = chocolates.loc[chocolates['dayofweek'] == 2]
disaggchocthu = chocolates.loc[chocolates['dayofweek'] == 3]
disaggchocfri = chocolates.loc[chocolates['dayofweek'] == 4]
disaggchocsun = chocolates.loc[chocolates['dayofweek'] == 6]

disaggchoctue = disaggchoctue.groupby('price ($)').agg({'Buy (1=buy, 0=not bu
y)': 'sum'})
disaggchocwed = disaggchocwed.groupby('price ($)').agg({'Buy (1=buy, 0=not bu
y)': 'sum'})
disaggchocthu = disaggchocthu.groupby('price ($)').agg({'Buy (1=buy, 0=not bu
y)': 'sum'})
disaggchocfri = disaggchocfri.groupby('price ($)').agg({'Buy (1=buy, 0=not bu
y)': 'sum'})
disaggchocsun = disaggchocsun.groupby('price ($)').agg({'Buy (1=buy, 0=not bu
y)': 'sum'})

disaggchoctue['price'] = disaggchoctue.index
disaggchocwed['price'] = disaggchocwed.index
disaggchocthu['price'] = disaggchocthu.index
disaggchocfri['price'] = disaggchocfri.index
disaggchocsun['price'] = disaggchocsun.index
```

In [15]:
```
disaggchocmon['revenue'] = disaggchocmon['Buy (1=buy, 0=not buy)'] * disaggcho
cmon['price']
disaggchocmon['cost'] = disaggchocmon['Buy (1=buy, 0=not buy)'] * 1
disaggchocmon['profit'] = disaggchocmon['revenue'] - disaggchocmon['cost']

disaggchocsat['revenue'] = disaggchocsat['Buy (1=buy, 0=not buy)'] * disaggcho
csat['price']
disaggchocsat['cost'] = disaggchocsat['Buy (1=buy, 0=not buy)'] * 1
disaggchocsat['profit'] = disaggchocsat['revenue'] - disaggchocsat['cost']

disaggchoctue['revenue'] = disaggchoctue['Buy (1=buy, 0=not buy)'] * disaggcho
ctue['price']
disaggchoctue['cost'] = disaggchoctue['Buy (1=buy, 0=not buy)'] * 1
disaggchoctue['profit'] = disaggchoctue['revenue'] - disaggchoctue['cost']

disaggchocwed['revenue'] = disaggchocwed['Buy (1=buy, 0=not buy)'] * disaggcho
cwed['price']
disaggchocwed['cost'] = disaggchocwed['Buy (1=buy, 0=not buy)'] * 1
disaggchocwed['profit'] = disaggchocwed['revenue'] - disaggchocwed['cost']

disaggchocthu['revenue'] = disaggchocthu['Buy (1=buy, 0=not buy)'] * disaggcho
cthu['price']
disaggchocthu['cost'] = disaggchocthu['Buy (1=buy, 0=not buy)'] * 1
disaggchocthu['profit'] = disaggchocthu['revenue'] - disaggchocthu['cost']

disaggchocfri['revenue'] = disaggchocfri['Buy (1=buy, 0=not buy)'] * disaggcho
cfri['price']
disaggchocfri['cost'] = disaggchocfri['Buy (1=buy, 0=not buy)'] * 1
disaggchocfri['profit'] = disaggchocfri['revenue'] - disaggchocfri['cost']

disaggchocsun['revenue'] = disaggchocsun['Buy (1=buy, 0=not buy)'] * disaggcho
csun['price']
disaggchocsun['cost'] = disaggchocsun['Buy (1=buy, 0=not buy)'] * 1
disaggchocsun['profit'] = disaggchocsun['revenue'] - disaggchocsun['cost']
```

In [16]:
```
disaggchocmon
```

Out[16]:

price ($)	Buy (1=buy, 0=not buy)	price	revenue	cost	profit
0.5	13	0.5	6.5	13	-6.5
0.6	3	0.6	1.8	3	-1.2
0.7	6	0.7	4.2	6	-1.8
0.8	8	0.8	6.4	8	-1.6
0.9	6	0.9	5.4	6	-0.6
1.0	8	1.0	8.0	8	0.0
1.1	6	1.1	6.6	6	0.6
1.2	4	1.2	4.8	4	0.8
1.3	8	1.3	10.4	8	2.4
1.4	6	1.4	8.4	6	2.4
1.5	4	1.5	6.0	4	2.0
1.6	2	1.6	3.2	2	1.2
1.7	2	1.7	3.4	2	1.4
1.8	2	1.8	3.6	2	1.6
1.9	4	1.9	7.6	4	3.6
2.0	0	2.0	0.0	0	0.0
2.1	0	2.1	0.0	0	0.0
2.2	0	2.2	0.0	0	0.0
2.3	0	2.3	0.0	0	0.0
2.4	0	2.4	0.0	0	0.0

In [17]:
```
disaggchoctue
```

Out[17]:

price ($)	Buy (1=buy, 0=not buy)	price	revenue	cost	profit
0.5	10	0.5	5.0	10	-5.0
0.6	5	0.6	3.0	5	-2.0
0.7	8	0.7	5.6	8	-2.4
0.8	11	0.8	8.8	11	-2.2
0.9	8	0.9	7.2	8	-0.8
1.0	8	1.0	8.0	8	0.0
1.1	6	1.1	6.6	6	0.6
1.2	3	1.2	3.6	3	0.6
1.3	6	1.3	7.8	6	1.8
1.4	6	1.4	8.4	6	2.4
1.5	4	1.5	6.0	4	2.0
1.6	3	1.6	4.8	3	1.8
1.7	5	1.7	8.5	5	3.5
1.8	1	1.8	1.8	1	0.8
1.9	3	1.9	5.7	3	2.7
2.0	3	2.0	6.0	3	3.0
2.1	1	2.1	2.1	1	1.1
2.2	1	2.2	2.2	1	1.2
2.3	0	2.3	0.0	0	0.0
2.4	0	2.4	0.0	0	0.0

In [18]:
```
disaggchocwed
```

Out[18]:

price ($)	Buy (1=buy, 0=not buy)	price	revenue	cost	profit
0.5	6	0.5	3.000000	6	-3.000000
0.6	6	0.6	3.600000	6	-2.400000
0.7	10	0.7	7.000000	10	-3.000000
0.8	5	0.8	4.000000	5	-1.000000
0.9	7	0.9	6.300000	7	-0.700000
1.0	3	1.0	3.000000	3	0.000000
1.1	7	1.1	7.700000	7	0.700000
1.2	7	1.2	8.400000	7	1.400000
1.3	15	1.3	19.499999	15	4.499999
1.4	1	1.4	1.400000	1	0.400000
1.5	3	1.5	4.500000	3	1.500000
1.6	3	1.6	4.800000	3	1.800000
1.7	5	1.7	8.500000	5	3.500000
1.8	1	1.8	1.800000	1	0.800000
1.9	4	1.9	7.600000	4	3.600000
2.0	1	2.0	2.000000	1	1.000000
2.1	4	2.1	8.400000	4	4.400000
2.2	3	2.2	6.600000	3	3.600000
2.3	0	2.3	0.000000	0	0.000000
2.4	1	2.4	2.400000	1	1.400000

In [19]:
```
disaggchocthu
```

Out[19]:

price ($)	Buy (1=buy, 0=not buy)	price	revenue	cost	profit
0.5	7	0.5	3.5	7	-3.5
0.6	9	0.6	5.4	9	-3.6
0.7	11	0.7	7.7	11	-3.3
0.8	8	0.8	6.4	8	-1.6
0.9	6	0.9	5.4	6	-0.6
1.0	7	1.0	7.0	7	0.0
1.1	8	1.1	8.8	8	0.8
1.2	5	1.2	6.0	5	1.0
1.3	5	1.3	6.5	5	1.5
1.4	1	1.4	1.4	1	0.4
1.5	2	1.5	3.0	2	1.0
1.6	4	1.6	6.4	4	2.4
1.7	4	1.7	6.8	4	2.8
1.8	1	1.8	1.8	1	0.8
1.9	0	1.9	0.0	0	0.0
2.0	2	2.0	4.0	2	2.0
2.1	0	2.1	0.0	0	0.0
2.2	0	2.2	0.0	0	0.0
2.3	0	2.3	0.0	0	0.0
2.4	0	2.4	0.0	0	0.0

In [20]:
```
disaggchocfri
```

Out[20]:

price ($)	Buy (1=buy, 0=not buy)	price	revenue	cost	profit
0.5	7	0.5	3.5	7	-3.5
0.6	7	0.6	4.2	7	-2.8
0.7	6	0.7	4.2	6	-1.8
0.8	4	0.8	3.2	4	-0.8
0.9	7	0.9	6.3	7	-0.7
1.0	8	1.0	8.0	8	0.0
1.1	9	1.1	9.9	9	0.9
1.2	7	1.2	8.4	7	1.4
1.3	2	1.3	2.6	2	0.6
1.4	3	1.4	4.2	3	1.2
1.5	3	1.5	4.5	3	1.5
1.6	3	1.6	4.8	3	1.8
1.7	1	1.7	1.7	1	0.7
1.8	4	1.8	7.2	4	3.2
1.9	1	1.9	1.9	1	0.9
2.0	1	2.0	2.0	1	1.0
2.1	1	2.1	2.1	1	1.1
2.2	0	2.2	0.0	0	0.0
2.3	0	2.3	0.0	0	0.0
2.4	0	2.4	0.0	0	0.0

In [21]:
```
disaggchocsat
```

Out[21]:

price ($)	Buy (1=buy, 0=not buy)	price	revenue	cost	profit
0.5	10	0.5	5.0	10	-5.0
0.6	6	0.6	3.6	6	-2.4
0.7	10	0.7	7.0	10	-3.0
0.8	9	0.8	7.2	9	-1.8
0.9	10	0.9	9.0	10	-1.0
1.0	9	1.0	9.0	9	0.0
1.1	12	1.1	13.2	12	1.2
1.2	10	1.2	12.0	10	2.0
1.3	9	1.3	11.7	9	2.7
1.4	6	1.4	8.4	6	2.4
1.5	7	1.5	10.5	7	3.5
1.6	11	1.6	17.6	11	6.6
1.7	0	1.7	0.0	0	0.0
1.8	4	1.8	7.2	4	3.2
1.9	2	1.9	3.8	2	1.8
2.0	0	2.0	0.0	0	0.0
2.1	4	2.1	8.4	4	4.4
2.2	1	2.2	2.2	1	1.2
2.3	1	2.3	2.3	1	1.3
2.4	0	2.4	0.0	0	0.0

In [22]:
```
disaggchocsun
```

Out[22]:

price ($)	Buy (1=buy, 0=not buy)	price	revenue	cost	profit
0.5	7	0.5	3.5	7	-3.5
0.6	5	0.6	3.0	5	-2.0
0.7	7	0.7	4.9	7	-2.1
0.8	5	0.8	4.0	5	-1.0
0.9	3	0.9	2.7	3	-0.3
1.0	2	1.0	2.0	2	0.0
1.1	1	1.1	1.1	1	0.1
1.2	1	1.2	1.2	1	0.2
1.3	6	1.3	7.8	6	1.8
1.4	6	1.4	8.4	6	2.4
1.5	4	1.5	6.0	4	2.0
1.6	0	1.6	0.0	0	0.0
1.7	0	1.7	0.0	0	0.0
1.8	0	1.8	0.0	0	0.0
1.9	2	1.9	3.8	2	1.8
2.0	2	2.0	4.0	2	2.0
2.1	0	2.1	0.0	0	0.0
2.2	2	2.2	4.4	2	2.4
2.3	2	2.3	4.6	2	2.6
2.4	0	2.4	0.0	0	0.0

On Monday, I would suggest setting the price to $1.90. On Tuesday, I would suggest $1.70. On Wednesday, I would suggest $1.30. On Thursday, I would suggest $1.70. On Friday, I would suggest $1.80. On Saturday, I would suggest setting the price to $1.60. On Sunday, I would suggest $2.30. These are all the prices that produce the highest possible profits on each day.

5) (**Requires knowledge in econometrics, optional) Specify demand as a function of observable customer characteristics and estimate it.Make sure what demand model you are estimating.

Please see attached STATA logs. It appears that we can specify demand as a function of observable characteristics as follows:

Q = a - bP + U; where Q is quantity bought (identified as Buy1buy0notbuy), a is a constant, b measures price sensitivity. P is a randomly assigned price for Anne's product, and U is an error term. There are two different demand models that occur through this method: one for safari users and one for mobile users.

I estimate that a = 63.91997, and b = -31.30259. Interpreting this, for every dollar price is increased, 31.302 units less of Anne's product are bought. This implies that users very price sensitive since this is quite a decrease.

What is also interesting is that the t-score for the mobile user coefficient implies that it is not statistically significant at a 95% confidence level. With 19 degrees of freedom (number of observations is 20), this t-score of 0.36 corresponds to a p-value of 0.727, which is not significant at p < .05. (Assuming a one-tailed hypothesis). The Safari coefficient is also not statistically significant, since its t score is 0.93 and its p-value is 0.369, which is also larger than 0.05.

Since the mobile and safari coefficients are not statistically significantly different from zero, it appears that my demand model has only one significant coefficient: price sensitivity. Hence, this is why my aggregated model excludes variables for safari or mobile users.

6) (**Requires knowledge in econometrics, optional) Based on estimated demand, derive the optimal pricing strategy as a function of observable customer characteristics.What are the characteristics that should make Anne charge the highestprices? What are the characteristics that should make Anne charge the lowest prices?

Given that my only significant coefficient for the aggregated model was price sensitivity, I cannot answer this question for an aggregated model.

However, for individual decisions, I can model a pricing strategy as follows: First, I specify a demand model: Buy = a + bP + cX + u; where a is a constant, u is the error term, b measures price sensitivity, and c measures other factors related to buying (such as being a safari or mobile user).

Running the logistic regression, I yield the following result: a = 11.95399, b = -9.464326, and c has the following significant coefficients (at p = 0.05): saturday = 1.722478, sunday = 1.288354, safari = 2.277246, and mobile = 6.797639. Coefficents for Monday, Thursday, Tuesday, and Wednesday are excluded because they are not statistically signficantly different from zero. The coefficient for Friday was not calculated likely due to a user by me.

Characteristics that should make Anne charge the highest prices include if the day of week is Saturday or Sunday (since it appears that more of Anne's product is bought on those days), or if the customer is a mobile user (since it appears that this binary variable contributes significantly to Anne's sales). She should charge higher prices based on these characteristics since these seem to be robust categories of buyers who appear ready to buy the product and are not as sensitive to price since there is a positive effect on buying from one of these categories being true.

The characteristic that should make Anne charge lower prices is the price sensitivity coefficient. Since it is highly negative in the logit model, this implies that a generic customer to Anne's business is very price sensitive, meaning that Anne has to be careful with regards to setting price and she should err on the side of caution and charge lower prices more often than not.