1. What percent of customers responded (i.e. bought anything) from this catalog?

We observe from our tabulated dummy values that 2.5% of customers responded from the catalog.

```
[8]  data['buy_dummy']=pandas.get_dummies(data['buyer'])['yes']

[9]  mba263.tabulate(data['buy_dummy'])
```

	Name	Count	Frequency
0	0	94180	0.975443
1	1	2371	0.024557

2. Of those who bought, what was the average dollars ordered from this catalog? (Hint: use a groupby with buy/no buy variable)

The average dollars ordered from the catalog is 518.10 dollars.

```
[10] data[ ['totdol' , 'buy_dummy'] ].groupby('buy_dummy').mean()
```

buy_dummy	totdol
0	331.991155
1	518.108393

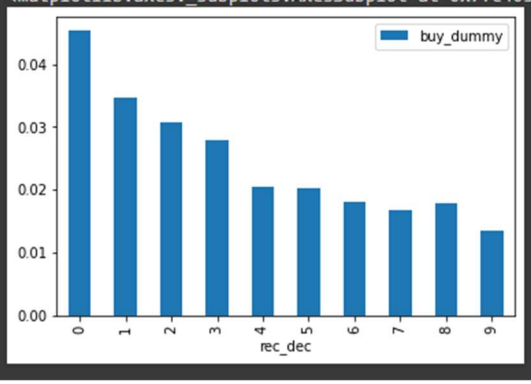
3. Create decile variables for recency, frequency and monetary. (Hint: use ntile command from mba263 module)

```
[18] data['rec_dec'] = mba263.ntile(data['last'],10)
     #subtract to reverse the order for frequency.
     data['freq_dec'] = 9 - mba263.ntile(data['numords'],10)
     #subtract to reverse the order for mv.
     data['mv_dec'] = 9 - mba263.ntile(data['totdol'],10)
```

4. Create a bar chart showing the response rate (i.e., the proportion of customers who bought something) to this catalog by recency decile.

```
data[ ['buy_dummy','rec_dec'] ].groupby('rec_dec').mean().plot(kind='bar')
```

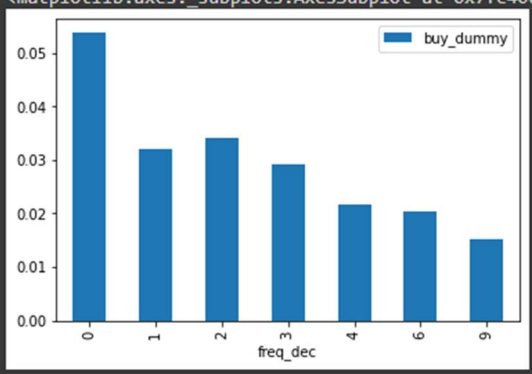
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc403da3cd0>
```



5. Create a bar chart showing the bar chart of response rate to this catalog by frequency decile. (Hint. use groupby with frequency deciles and plot command)

```
data[ ['buy_dummy','freq_dec'] ].groupby('freq_dec').mean().plot(kind='bar')
```

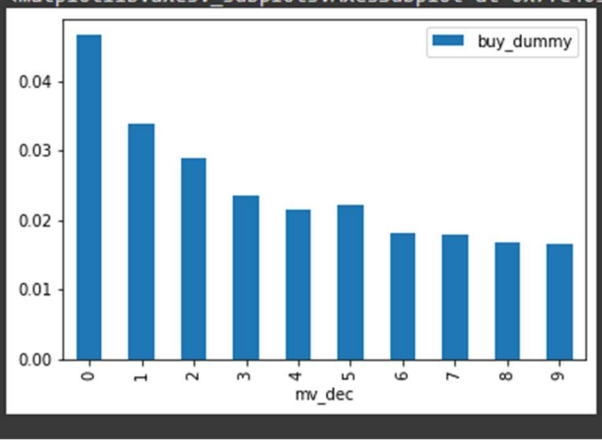
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc40623bfd0>
```



6. Create a bar chart showing the bar chart of response rate to this catalog by monetary decile.

```
data[ ['buy_dummy','mv_dec'] ].groupby('mv_dec').mean().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc40397a880>
```



7. Using only those customers who placed an order from this catalog, create three bar charts showing the average dollars ordered from this catalog by recency, frequency and monetary deciles, respectively.(Hint: first create a data frame with those rows that ordered, and use groupby(...).mean().plot(...) on the restricted data)

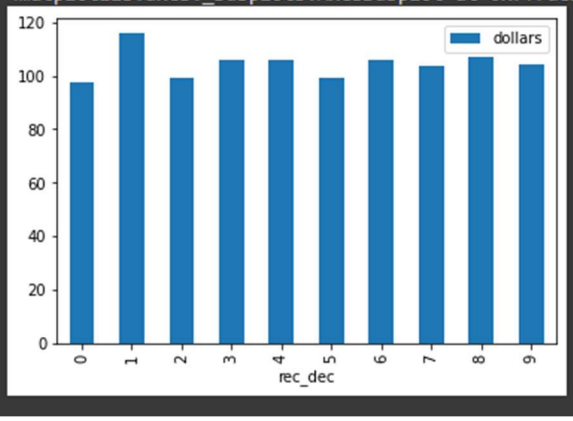
Assumption: Used dollars column/feature since prompt asks for average dollars and not total.

Average dollars by recency deciles

```
responded = data[data['buyer'] == 'yes']
```

```
[16] responded[['dollars', 'rec_dec']].groupby('rec_dec').mean().plot(kind='bar')
```

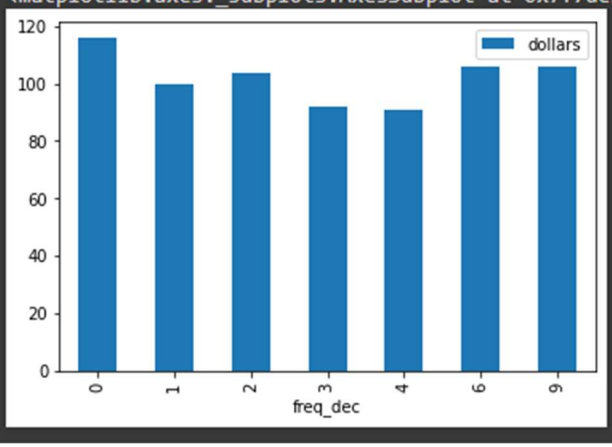
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7de9d2e580>
```



Average dollars by frequency deciles

```
[17] responded[['dollars', 'freq_dec']].groupby('freq_dec').mean().plot(kind='bar')
```

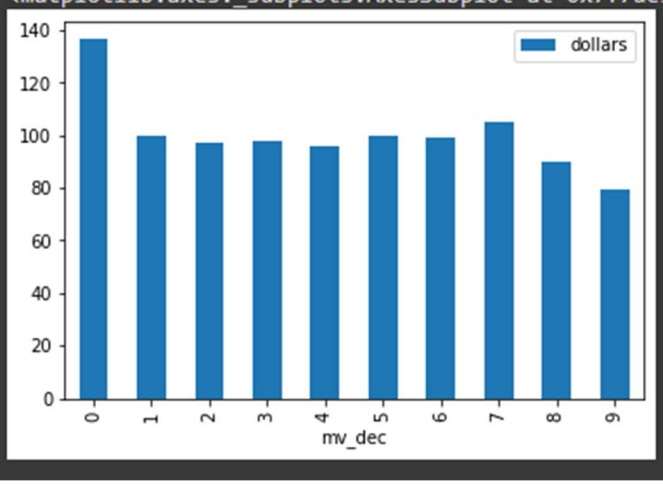
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7de9cb0820>
```



Average dollars by monetary deciles

```
[13] responded[['dollars', 'mv_dec']].groupby('mv_dec').mean().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f7de9c7cdc0>
```

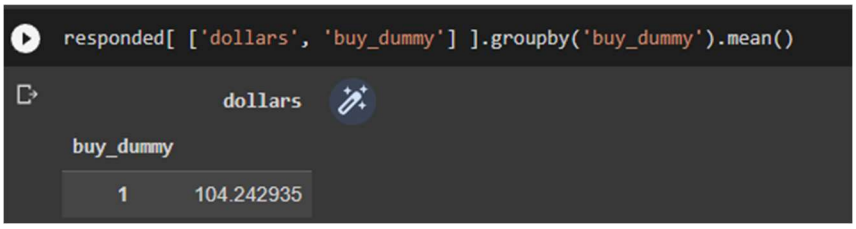


8. What do the above bar charts reveal about the likelihood of response and the size of the order across the different recency, frequency, and monetary deciles?

There is not a lot of variances in the average dollar amount spent for recency and frequency charts but we can observe that that dollar amount is expected to be higher for the monetary charts since the more frequently customer spends, the higher the dollar amount will be.

The second observation is that the frequency chart is missing several bins and contains a few segments from the top and few from the bottom categories whereas recency and monetary charts have all 10 bins.

9. Consider mailing the catalog to the remaining 1,834,469 (=1,931,020-96,551) customers.
Calculate
(a) the expected gross profit in dollars,
(b) the expected gross profit as a % of expected gross sales, and
(c) the expected return on marketing expenditures (gross profit/cost to mail catalogs) To calculate (a),(b), and (c) assume:
Cost to produce and mail catalog = $1
COGS and variable costs on orders = 50% (of sales revenues)



Assumption: From question 1 only 2371 people responded 96551 did not, therefore we can assume that (1,834,469/96551) * 2371 = 45049 will be the remaining people to respond.

a) Expected gross profit = (45049 * 104.2) - (0.5 * 45049 * 104.2) - (1,834,469 * 1) = 2,347,953 - 1,834,469 = **$513,484**

b) Expected gross profit as % of expected sales = (513,484 * 100) / (45049*104.2) = **10.9%**

c) Expected return on Marketing expenditures = (513,484 * 100) / 1,834,469 = **28%**

10. What is the breakeven response rate?

Average customer spends 104.2 dollars for every catalog sent, profit is 50% of 104.2 which is $52.1 and the cost to send the catalog is $1.

Therefore, the breakeven response rate = $1 * 100/($104.2 * 50%) = **1.92%**

11. Consider mailing only to those of the remaining 1,834,469 customers in RFM cells (using the sequential n-tiles approach) with a response rate that is equal to or greater than the breakeven response rate. Specifically, compute the following items. You can do this by following the steps described in RFM jupyter notebook and using a calculator:
• Determine which RFM segments (using the sequential n-tiles approach) have response rates exceeding the breakeven rate.
• Determine the number of customers belonging to these profitable segments.
• Determine the number of buyers belonging to these profitable segments.
• Finally, what would the
(a) the expected gross profit in dollars,
(b) the expected gross profit as a % of expected gross sales, and
(c) the expected return on marketing expenditures (gross profit/cost to mail catalogs)
have been as a result of mailing the catalog only to those customers in the RFM cells with response rates exceeding the breakeven? That is, rather than mailing to all

1,834,469 customers – what would the profitability of the mailing have been if mailed to the subset of customers in 'profitable' segments?

```
[24] data['responserate_sq'] = mba263.get_means(data['buy_dummy'], data['rfm1'])

[25] data['mailto_rfm'] = (data['responserate_sq'] > 0.0192) * 1

[26] rfm1_mailed = data[data['mailto_rfm'] == 1]

[27] mba263.tabulate(rfm1_mailed['buy_dummy'])
```

	Name	Count	Frequency
0	0	50532	0.966398
1	1	1757	0.033602

Customers in profitable segment = (50532 + 1757) * 19 = 993,491

Buyers in profitable segment = 1757 * 19 = 33,383

a) Expected gross profit in dollars (33,383 * 104.2) - (0.5 * 33,383 * 104.2) - (993,491 * 1)= **$746,597**

b) Expected gross profit as % of expected sales = (746,597 * 100)/(33,383*104.2)= **21.4%**

c) Expected return on Marketing expenditures = (746,597 * 100) / 993,491= **75.1%**

12. Examine the first 20 or so observations in the database. What do you notice about the RFM1 and RFM2 values? That is – do the two approaches generally yield the same RFM index for any given customer? What do you see as the pros and cons of the two approaches (from a statistical as well as logical perspective) and why?

Looking at the first 20 or so observations in the database, we can see that the RFM1 and RFM2 values for recency are the same. The two approaches generally yield the same RFM index for any given customer because they both group customers into different RFM groups which can help decision-makers develop more effective marketing and sale strategies for targeting buying customers which will improve profitability.

The cons of sequential-n-tile method is that the frequency in RFM may be hard to interpret, the pro of using sequential-n-tile method is that it can result in a smoother distribution of RFM score which helps marketers get more segmentation to get better accuracy and target population to advertise to.

The cons of independent-n-tile is that it may not create enough uniform distributions of RFM scores. The pro of independent-n-tile is that it makes it very easy to interpret the RFM scores as recency, frequency and monetary rank are assigned independently.