In [91]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
/usr/local/lib/python3.6/dist-packages/statsmodels/tools/_testing.py:19: F
utureWarning: pandas.util.testing is deprecated. Use the functions in the
public API at pandas.testing instead.
  import pandas.util.testing as tm
```

In [36]:

```
from google.colab import files
uploaded = files.upload()
```

Choose Files No file chosen

Upload widget is only available when the cell has been executed in the current browser session.
Please rerun this cell to enable.

```
Saving ANZ.csv to ANZ (1).csv
```

In [ ]:

```
import io
df = pd.read_csv(io.BytesIO(uploaded['ANZ.csv']))
```

In [65]:

```
df.head()
```

Out[65]:

	status	card_present_flag	bpay_biller_code	account	currency	long_lat	txn_descri
0	authorized	1.0	NaN	ACC-1598451071	AUD	153.41 -27.95	
1	authorized	0.0	NaN	ACC-1598451071	AUD	153.41 -27.95	SALES
2	authorized	1.0	NaN	ACC-1222300524	AUD	151.23 -33.94	
3	authorized	1.0	NaN	ACC-1037050564	AUD	153.10 -27.66	SALES
4	authorized	1.0	NaN	ACC-1598451071	AUD	153.41 -27.95	SALES

In [66]:

```
df.describe()
```

Out[66]:

	card_present_flag	merchant_code	balance	age	amount
count	7717.000000	883.0	12043.000000	12043.000000	12043.000000
mean	0.802644	0.0	14704.195553	30.582330	187.933588
std	0.398029	0.0	31503.722652	10.046343	592.599934
min	0.000000	0.0	0.240000	18.000000	0.100000
25%	1.000000	0.0	3158.585000	22.000000	16.000000
50%	1.000000	0.0	6432.010000	28.000000	29.000000
75%	1.000000	0.0	12465.945000	38.000000	53.655000
max	1.000000	0.0	267128.520000	78.000000	8835.980000

In [67]:

```
df.count()
```

Out[67]:

```
status              12043
card_present_flag    7717
bpay_biller_code      885
account             12043
currency            12043
long_lat            12043
txn_description     12043
merchant_id          7717
merchant_code         883
first_name          12043
balance             12043
date                12043
gender              12043
age                 12043
merchant_suburb      7717
merchant_state       7717
extraction          12043
amount              12043
transaction_id      12043
country             12043
customer_id         12043
merchant_long_lat    7717
movement            12043
dtype: int64
```

In [68]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12043 entries, 0 to 12042
Data columns (total 23 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   status             12043 non-null  object 
 1   card_present_flag  7717 non-null   float64
 2   bpay_biller_code   885 non-null    object 
 3   account            12043 non-null  object 
 4   currency           12043 non-null  object 
 5   long_lat           12043 non-null  object 
 6   txn_description    12043 non-null  object 
 7   merchant_id        7717 non-null   object 
 8   merchant_code      883 non-null    float64
 9   first_name         12043 non-null  object 
 10  balance            12043 non-null  float64
 11  date               12043 non-null  object 
 12  gender             12043 non-null  object 
 13  age                12043 non-null  int64  
 14  merchant_suburb    7717 non-null   object 
 15  merchant_state     7717 non-null   object 
 16  extraction         12043 non-null  object 
 17  amount             12043 non-null  float64
 18  transaction_id     12043 non-null  object 
 19  country            12043 non-null  object 
 20  customer_id        12043 non-null  object 
 21  merchant_long_lat  7717 non-null   object 
 22  movement           12043 non-null  object 
dtypes: float64(4), int64(1), object(18)
memory usage: 2.1+ MB
```

In [69]:

```
df['date']
```

Out[69]:

```
0        01-08-18
1        01-08-18
2        01-08-18
3        01-08-18
4        01-08-18
           ...
12038    31-10-18
12039    31-10-18
12040    31-10-18
12041    31-10-18
12042    31-10-18
Name: date, Length: 12043, dtype: object
```

In [ ]:

```
df['date']= pd.to_datetime(df['date'])
```

In [71]:

```
df['date']
```

Out[71]:

```
0       2018-01-08
1       2018-01-08
2       2018-01-08
3       2018-01-08
4       2018-01-08
           ...
12038   2018-10-31
12039   2018-10-31
12040   2018-10-31
12041   2018-10-31
12042   2018-10-31
Name: date, Length: 12043, dtype: datetime64[ns]
```

In [72]:

```
df.head()
```

Out[72]:

	status	card_present_flag	bpay_biller_code	account	currency	long_lat	txn_descri
0	authorized	1.0	NaN	ACC-1598451071	AUD	153.41 -27.95	
1	authorized	0.0	NaN	ACC-1598451071	AUD	153.41 -27.95	SALES
2	authorized	1.0	NaN	ACC-1222300524	AUD	151.23 -33.94	
3	authorized	1.0	NaN	ACC-1037050564	AUD	153.10 -27.66	SALES
4	authorized	1.0	NaN	ACC-1598451071	AUD	153.41 -27.95	SALES

In [ ]:

```
#some satatistics
median_amt=df['amount'].median()
max_amt=df['amount'].max()
min_amt=df['amount'].min()
mean_amt=df['amount'].mean()
```

In [75]:

```
print("Total number of transactions is ",str(df.status.count()))
print('The median amount is ',str(median_amt))
print('The mean amount is ',str(mean_amt))
print('The maximum transaction amount is ',str(max_amt))
print('The minimum transaction amount is ',str(min_amt))
```

```
Total number of transactions is  12043
The median amount is  29.0
The mean amount is  187.93358797641767
The maximum transaction amount is  8835.98
The minimum transaction amount is  0.1
```

In [ ]:

```
median_age=df['age'].median()
max_age=df['age'].max()
min_age=df['age'].min()
mean_age=df['age'].mean()
```

In [80]:

```
print("The age range is ",str(max_age-min_age))
print('The median age is ',str(median_age))
print('The mean age  is ',str(mean_age))
print('The maximum  age is ',str(max_age))
print('The minimum  age is ',str(min_age))
```

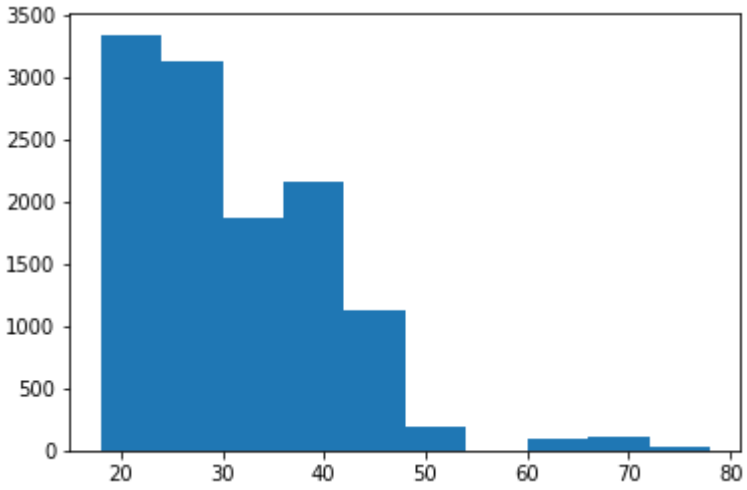
```
The age range is  60
The median age is  28.0
The mean age  is  30.5823299842232
The maximum  age is  78
The minimum  age is  18
```

In [82]:

```
plt.hist(df['age'])
```

Out[82]:

```
(array([3341., 3131., 1874., 2151., 1128.,  194.,    0.,   88.,  102.,
          34.]),
 array([18., 24., 30., 36., 42., 48., 54., 60., 66., 72., 78.]),
 <a list of 10 Patch objects>)
```

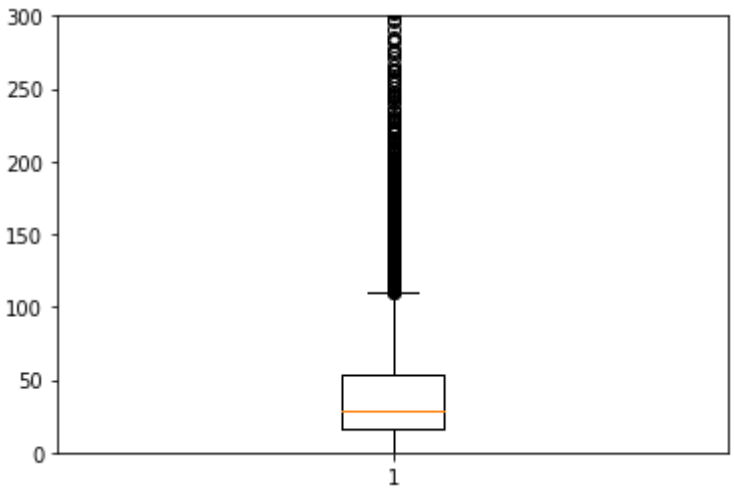


In [86]:

```
plt.boxplot(df['amount'])
plt.ylim(0,300)
df['amount'].describe()
```

Out[86]:

```
count    12043.000000
mean       187.933588
std        592.599934
min          0.100000
25%         16.000000
50%         29.000000
75%         53.655000
max       8835.980000
Name: amount, dtype: float64
```

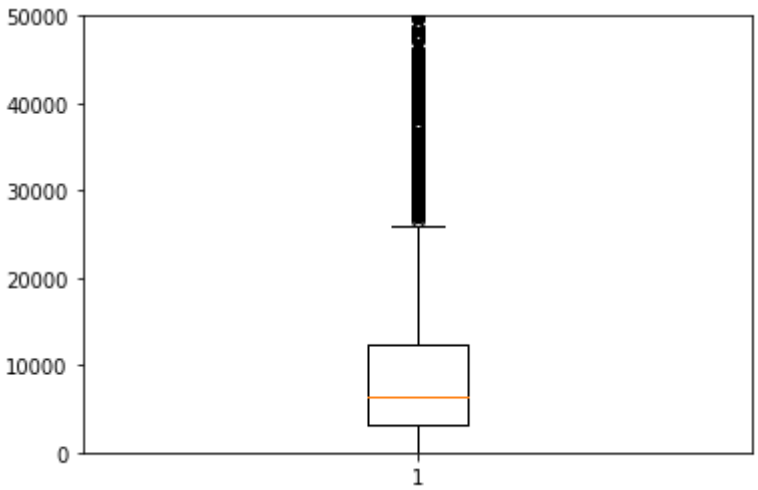


In [90]:

```
plt.boxplot(df['balance'])
plt.ylim(0,50000)
df['balance'].describe()
```

Out[90]:

```
count     12043.000000
mean      14704.195553
std       31503.722652
min           0.240000
25%        3158.585000
50%        6432.010000
75%       12465.945000
max      267128.520000
Name: balance, dtype: float64
```

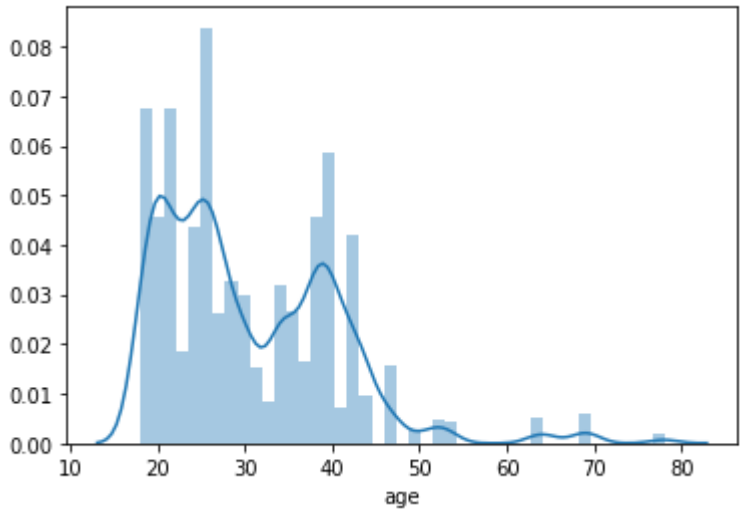


In [93]:

```
sns.distplot(df['age'])
```

Out[93]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcfdc246128>
```

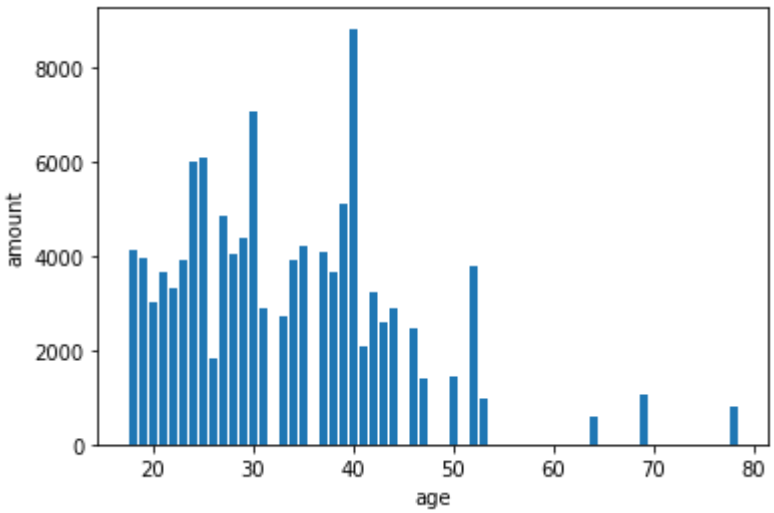


In [96]:

```
plt.bar(df['age'],df['amount'])
plt.xlabel("age")
plt.ylabel("amount")
```

Out[96]:

```
Text(0, 0.5, 'amount')
```

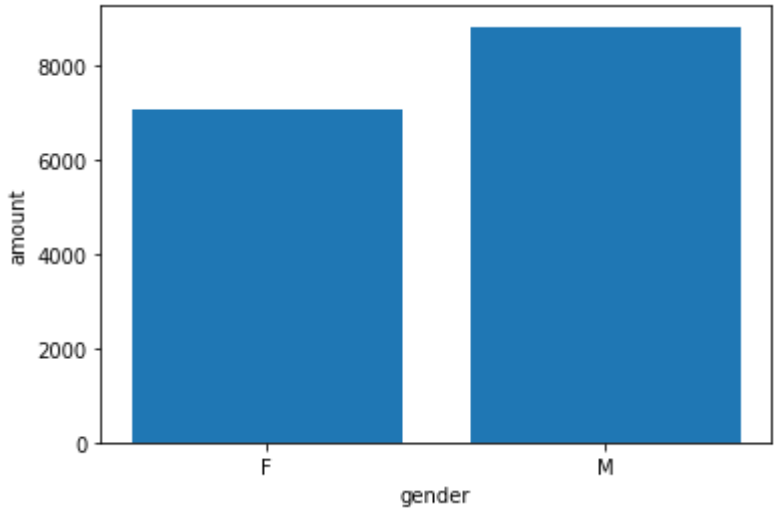


In [103]:

```
plt.bar(df['gender'],df['amount'])
plt.xlabel("gender")
plt.ylabel("amount")
df['gender'].value_counts()
```

Out[103]:

```
M    6285
F    5758
Name: gender, dtype: int64
```



In [ ]:

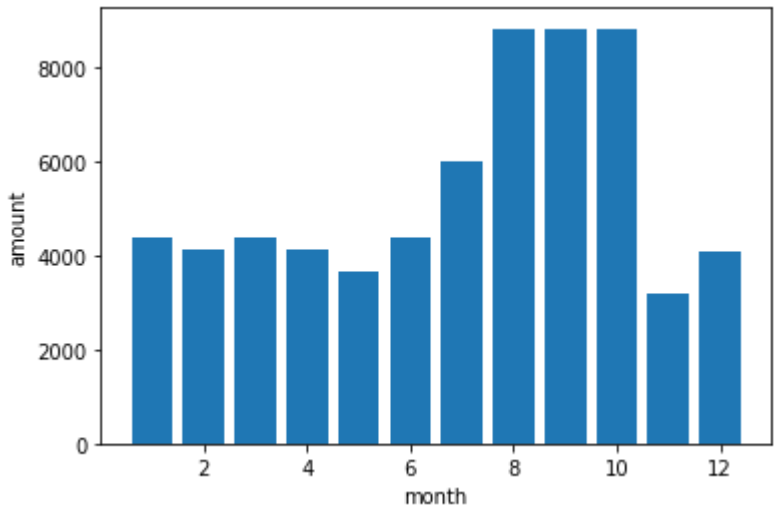
```
df['month'] = pd.DatetimeIndex(df['date']).month
```

In [104]:

```
plt.bar(df['month'],df['amount'])
plt.xlabel("month")
plt.ylabel("amount")
df['month'].value_counts()
```

Out[104]:

```
10    2885
9     2823
8     2750
3      426
5      417
12     412
2      405
4      402
11     394
6      381
1      377
7      371
Name: month, dtype: int64
```



In [ ]:

```

```