In [10]:

```
import pandas as pd
import numpy as np
import seaborn as sns
df=pd.read_csv("transfusion.csv")
df
```

Out[10]:

| | Recency (months) | Frequency (times) | Monetary (c.c. blood) | Time (months) | whether he/she donated blood in March 2007 |
|---|---|---|---|---|---|
| **0** | 2 | 50 | 12500 | 98 | 1 |
| **1** | 0 | 13 | 3250 | 28 | 1 |
| **2** | 1 | 16 | 4000 | 35 | 1 |
| **3** | 2 | 20 | 5000 | 45 | 1 |
| **4** | 1 | 24 | 6000 | 77 | 0 |
| **...** | ... | ... | ... | ... | ... |
| **743** | 23 | 2 | 500 | 38 | 0 |
| **744** | 21 | 2 | 500 | 52 | 0 |
| **745** | 23 | 3 | 750 | 62 | 0 |
| **746** | 39 | 1 | 250 | 39 | 0 |
| **747** | 72 | 1 | 250 | 72 | 0 |

748 rows × 5 columns

In [11]:

```
df.head()
```

Out[11]:

| | Recency (months) | Frequency (times) | Monetary (c.c. blood) | Time (months) | whether he/she donated blood in March 2007 |
|---|---|---|---|---|---|
| **0** | 2 | 50 | 12500 | 98 | 1 |
| **1** | 0 | 13 | 3250 | 28 | 1 |
| **2** | 1 | 16 | 4000 | 35 | 1 |
| **3** | 2 | 20 | 5000 | 45 | 1 |
| **4** | 1 | 24 | 6000 | 77 | 0 |

In [12]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 748 entries, 0 to 747
Data columns (total 5 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   Recency (months)                            748 non-null    int64
 1   Frequency (times)                           748 non-null    int64
 2   Monetary (c.c. blood)                       748 non-null    int64
 3   Time (months)                               748 non-null    int64
 4   whether he/she donated blood in March 2007  748 non-null    int64
dtypes: int64(5)
memory usage: 29.3 KB
```

In [13]:

```
df.isnull().sum()
```

```
Out[13]: Recency (months)                              0
         Frequency (times)                             0
         Monetary (c.c. blood)                         0
         Time (months)                                 0
         whether he/she donated blood in March 2007    0
         dtype: int64
```

```
In [14]: df.shape
```

```
Out[14]: (748, 5)
```

```
In [15]: df.describe
```

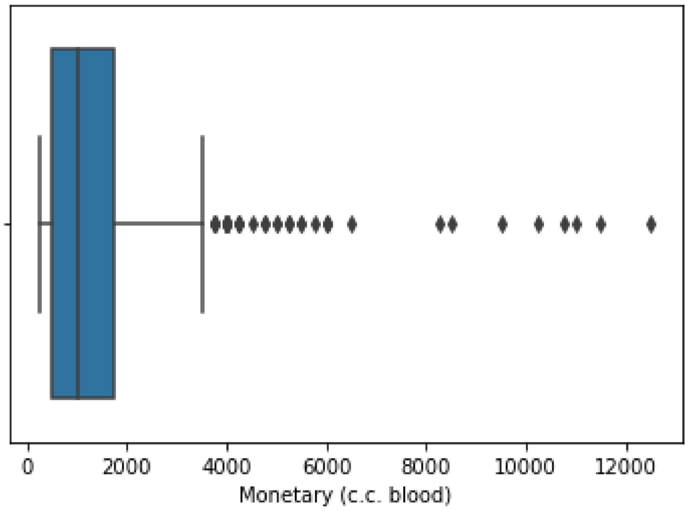
```
Out[15]: <bound method NDFrame.describe of      Recency (months)  Frequency (times)  Moneta
         ry (c.c. blood)  \
         0                   2                 50                  12500
         1                   0                 13                   3250
         2                   1                 16                   4000
         3                   2                 20                   5000
         4                   1                 24                   6000
         ..                ...                ...                    ...
         743                23                  2                    500
         744                21                  2                    500
         745                23                  3                    750
         746                39                  1                    250
         747                72                  1                    250

              Time (months)  whether he/she donated blood in March 2007
         0               98                                           1
         1               28                                           1
         2               35                                           1
         3               45                                           1
         4               77                                           0
         ..             ...                                         ...
         743             38                                           0
         744             52                                           0
         745             62                                           0
         746             39                                           0
         747             72                                           0

         [748 rows x 5 columns]>
```

```
In [19]: sns.boxplot(x="Monetary (c.c. blood)",data=df)
```

```
Out[19]: <AxesSubplot:xlabel='Monetary (c.c. blood)'>
```



```
In [33]: q1=df['Monetary (c.c. blood)'].quantile(0.25)
```

```
q3=df['Monetary (c.c. blood)'].quantile(0.75)
print("First Quantile=",q1,"\nSecond Quantile =",q3)
```

```
First Quantile= 500.0
Second Quantile = 1750.0
```

In [34]:

```
IQR=q3-q1
ul=q3+1.5*IQR
ll=q1-1.5*IQR
df=df[((df['Monetary (c.c. blood)']>=ll)&(df['Monetary (c.c. blood)']<=ul))]
df
```

Out[34]:

| | Recency (months) | Frequency (times) | Monetary (c.c. blood) | Time (months) | whether he/she donated blood in March 2007 |
|---|---|---|---|---|---|
| **1** | 0 | 13 | 3250 | 28 | 1 |
| **5** | 4 | 4 | 1000 | 4 | 0 |
| **6** | 2 | 7 | 1750 | 14 | 1 |
| **7** | 1 | 12 | 3000 | 35 | 0 |
| **8** | 2 | 9 | 2250 | 22 | 1 |
| **...** | ... | ... | ... | ... | ... |
| **743** | 23 | 2 | 500 | 38 | 0 |
| **744** | 21 | 2 | 500 | 52 | 0 |
| **745** | 23 | 3 | 750 | 62 | 0 |
| **746** | 39 | 1 | 250 | 39 | 0 |
| **747** | 72 | 1 | 250 | 72 | 0 |

703 rows × 5 columns

In [35]:

```
df.shape
```

Out[35]:

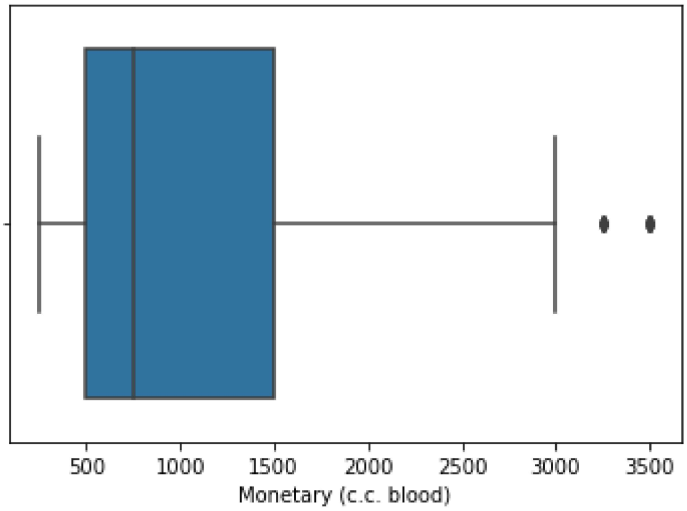
```
(703, 5)
```

In [36]:

```
sns.boxplot(x="Monetary (c.c. blood)",data=df)
```

Out[36]:

```
<AxesSubplot:xlabel='Monetary (c.c. blood)'>
```



In [37]:

```
from scipy import stats
```

In [38]:
```
z=np.abs(stats.zscore(df['Monetary (c.c. blood)']))
```

In [39]:
```
df=df[(z<3)]
```

In [40]:
```
df
```

Out[40]:

| | Recency (months) | Frequency (times) | Monetary (c.c. blood) | Time (months) | whether he/she donated blood in March 2007 |
|---|---|---|---|---|---|
| **1** | 0 | 13 | 3250 | 28 | 1 |
| **5** | 4 | 4 | 1000 | 4 | 0 |
| **6** | 2 | 7 | 1750 | 14 | 1 |
| **7** | 1 | 12 | 3000 | 35 | 0 |
| **8** | 2 | 9 | 2250 | 22 | 1 |
| **...** | ... | ... | ... | ... | ... |
| **743** | 23 | 2 | 500 | 38 | 0 |
| **744** | 21 | 2 | 500 | 52 | 0 |
| **745** | 23 | 3 | 750 | 62 | 0 |
| **746** | 39 | 1 | 250 | 39 | 0 |
| **747** | 72 | 1 | 250 | 72 | 0 |

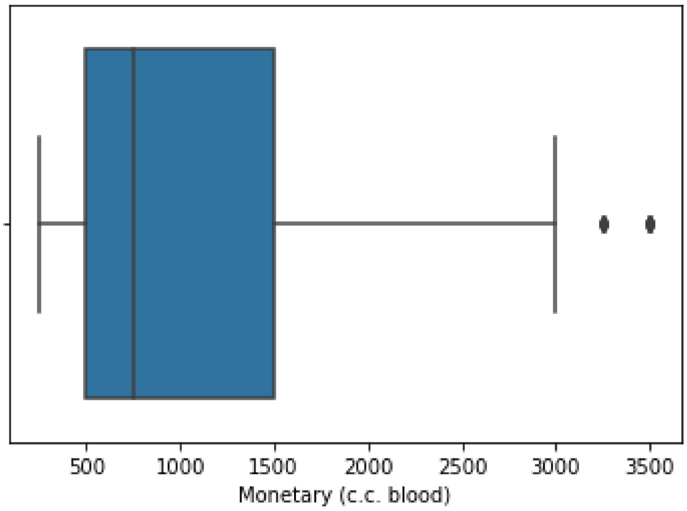
703 rows × 5 columns

In [41]:
```
print(df.shape)
```

```
(703, 5)
```

In [42]:
```
sns.boxplot(x="Monetary (c.c. blood)",data=df)
```

Out[42]:
```
<AxesSubplot:xlabel='Monetary (c.c. blood)'>
```



In [5]:
```
from sklearn.ensemble import IsolationForest
model=IsolationForest(n_estimators=50, max_samples='auto', contamination=float(0.1
model.fit('Monetary (c.c. blood)[['weights(in Kg)']])
Monetary (c.c. blood)['scores']=model.decision_function(Monetary (c.c. blood)[['we
Monetary (c.c. blood)['anomaly']=model.predict(Monetary (c.c. blood)[['weights(in 
anomaly=Monetary (c.c. blood).loc[Monetary (c.c. blood)['anomaly']==-1]
```

```
anomaly_index=list(anomaly.index)
print(anomaly)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [5], in <cell line: 3>()
      1 from sklearn.ensemble import IsolationForest
      2 model=IsolationForest(n_estimators=50, max_samples='auto', contamination=f
loat(0.1),max_features=1.0)
----> 3 model.fit(Monetary (c.c. blood)[['weights(in Kg)']])
      4 Monetary (c.c. blood)['scores']=model.decision_function(Monetary (c.c. blo
od)[['weights(in Kg)']])
      5 Monetary (c.c. blood)['anomaly']=model.predict(Monetary (c.c. blood)[['wei
ghts(in Kg)']])

NameError: name 'Monetary' is not defined
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: