Liver Disorder Dataset

In [ ]:

```
'''
1. Title: BUPA liver disorders

2. Source information:
   -- Creators: BUPA Medical Research Ltd.
   -- Donor: Richard S. Forsyth
             8 Grosvenor Avenue
             Mapperley Park
             Nottingham NG3 5DX
             0602-621676
   -- Date: 5/15/1990

3. Past usage:
   -- None known other than what is shown in the PC/BEAGLE User's Guide
      (written by Richard S. Forsyth).

4. Relevant information:
   -- The first 5 variables are all blood tests which are thought
      to be sensitive to liver disorders that might arise from
      excessive alcohol consumption.  Each line in the bupa.data file
      constitutes the record of a single male individual.
   -- It appears that drinks>5 is some sort of a selector on this database.
      See the PC/BEAGLE User's Guide for more information.

5. Number of instances: 345

6. Number of attributes: 7 overall

7. Attribute information:
   1. mcv       mean corpuscular volume
   2. alkphos   alkaline phosphotase
   3. sgpt      alamine aminotransferase
   4. sgot      aspartate aminotransferase
   5. gammagt   gamma-glutamyl transpeptidase
   6. drinks    number of half-pint equivalents of alcoholic beverages
                drunk per day
   7. selector  field used to split data into two sets

8. Missing values: none
'''
```

In [1]:

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

#Load Bupa.csv into a pandas dataFrame.
bupa = pd.read_csv("Bupa.csv")
```

In [2]:

```
# (Q) how many data-points and featrues are there?
print (bupa.shape)
```

```
(345, 7)
```

In [3]:

```
#(Q) What are the column names in our dataset?
print (bupa.columns)
```

```
Index([u'mcv', u'alkphos', u'sgpt', u'sgot', u'gammagt', u'drinks',
       u'selector'],
      dtype='object')
```

In [4]:

```
#(Q) How many data points for each class are present?
#(or) How many flowers for each species are present?

bupa["selector"].value_counts()
# balanced-dataset vs imbalanced datasets
#bupa is a nearly balanced dataset as the number of data points for 1 class is 145 and
 for 2 is 200.
```

Out[4]:

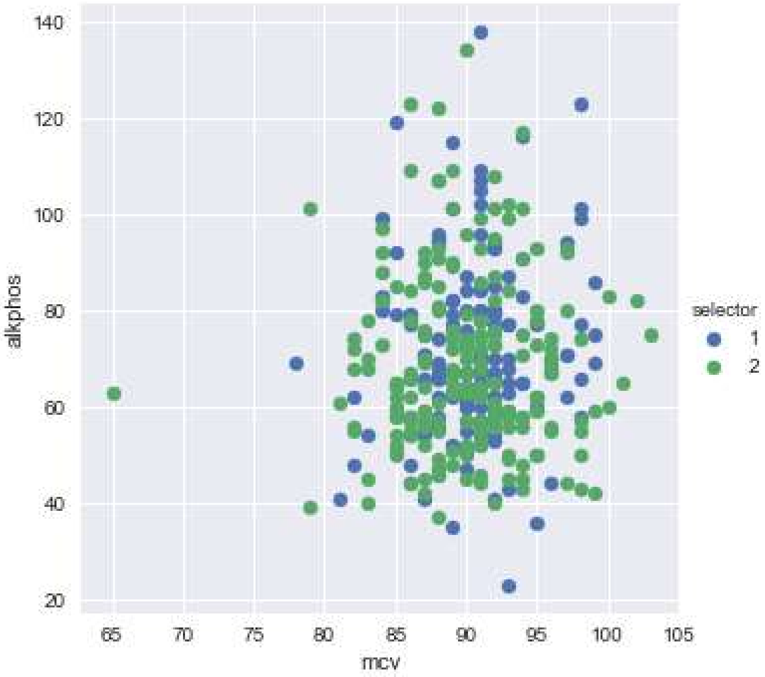
```
2    200
1    145
Name: selector, dtype: int64
```

2D Scatter plot

In [5]:

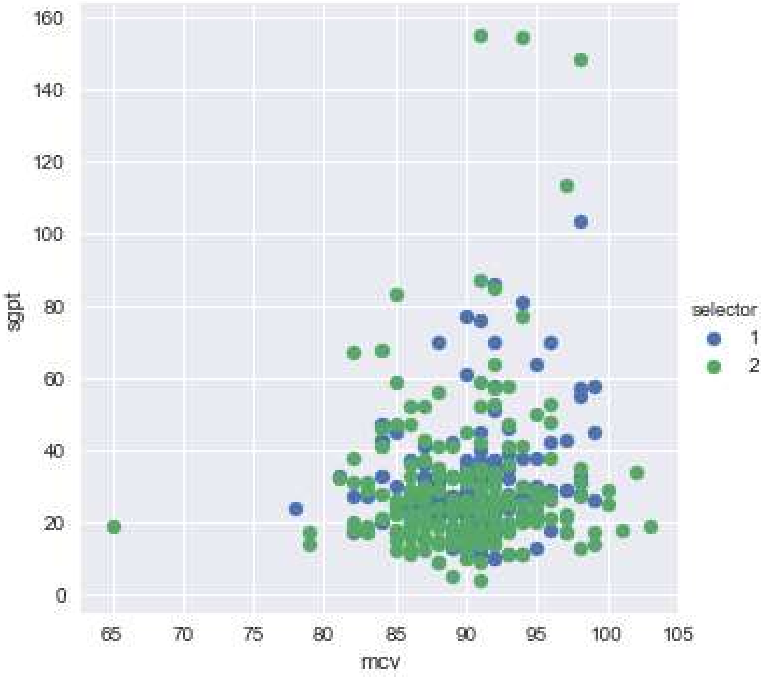
```
# 2-D Scatter plot with color-coding for each class.
# Here 'sns' corresponds to seaborn.
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "mcv", "alkphos") \
   .add_legend();
plt.show();

# Notice that the blue points can be easily seperated from red and green by drawing a l
ine.
# But read and green data points cannot be easily seperated.
# Can we draw multiple 2-D scatter plots for each combination of features?
# How many cobinations exist? 6C2 = 15.
```



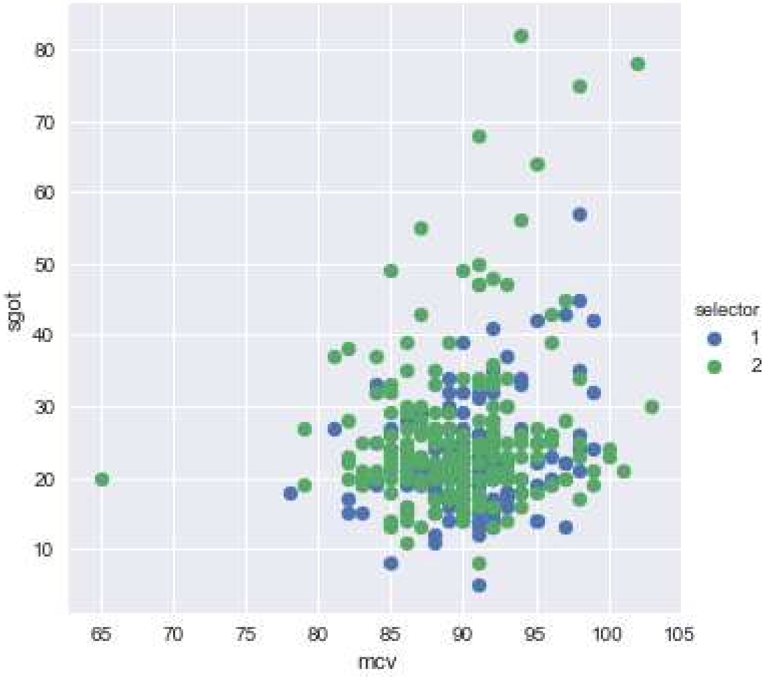
In [6]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "mcv", "sgpt") \
   .add_legend();
plt.show();
```



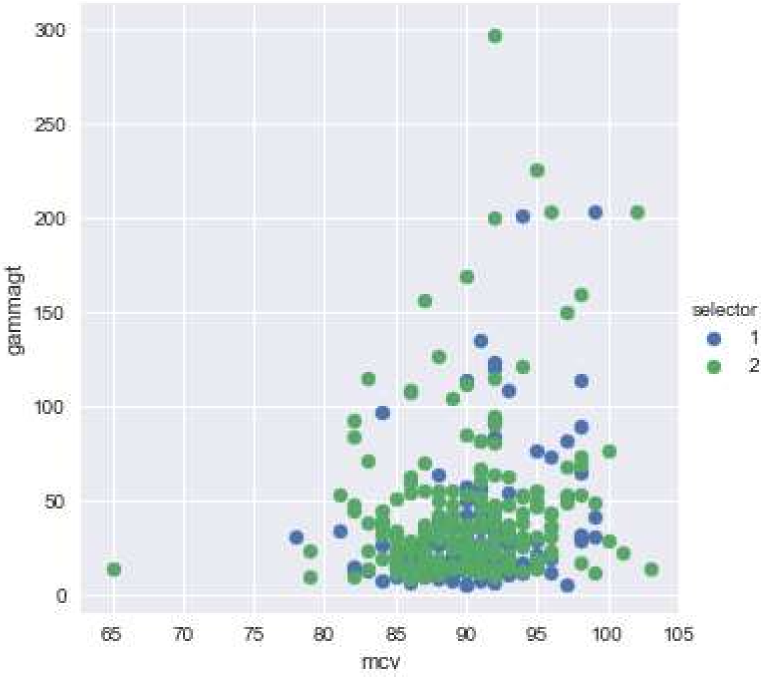
In [7]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "mcv", "sgot") \
   .add_legend();
plt.show();
```



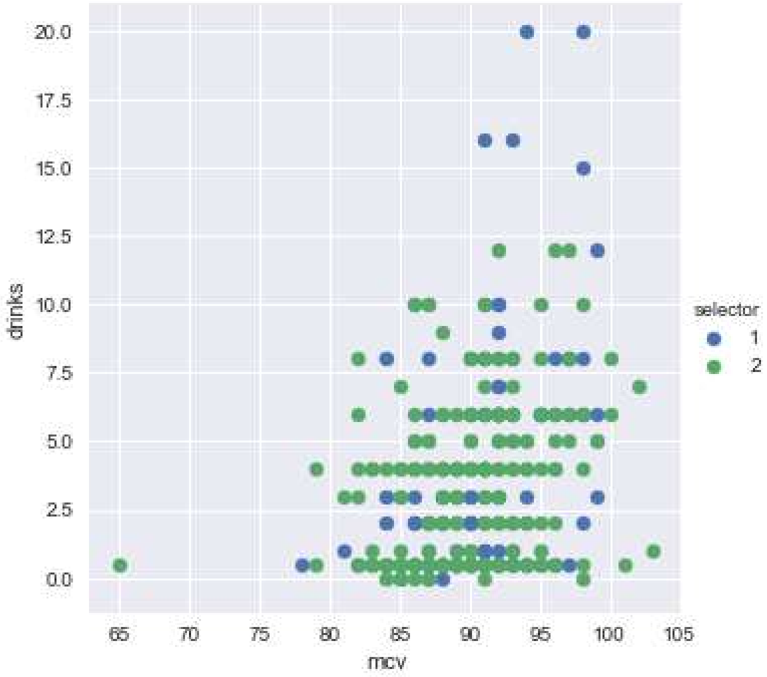
In [8]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "mcv", "gammagt") \
   .add_legend();
plt.show();
```



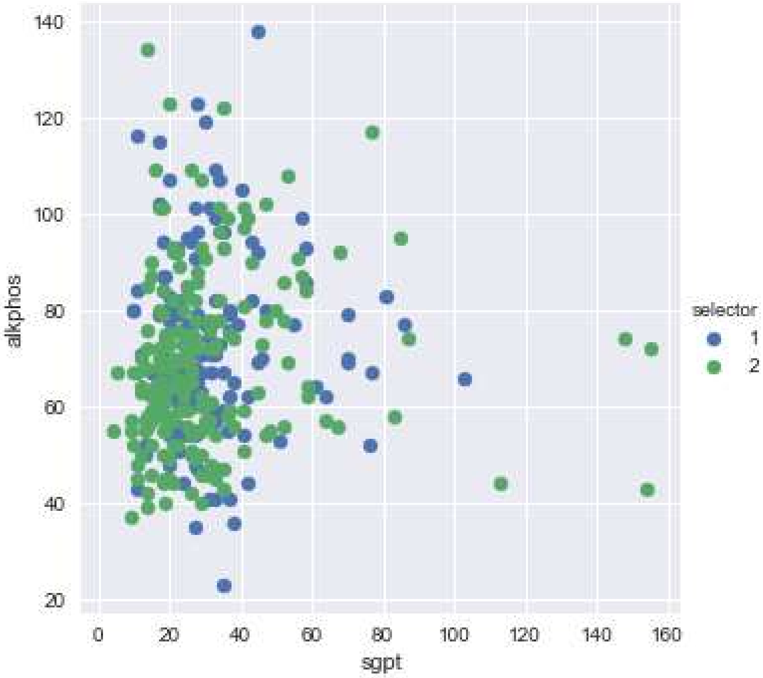
In [9]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "mcv", "drinks") \
   .add_legend();
plt.show();
```



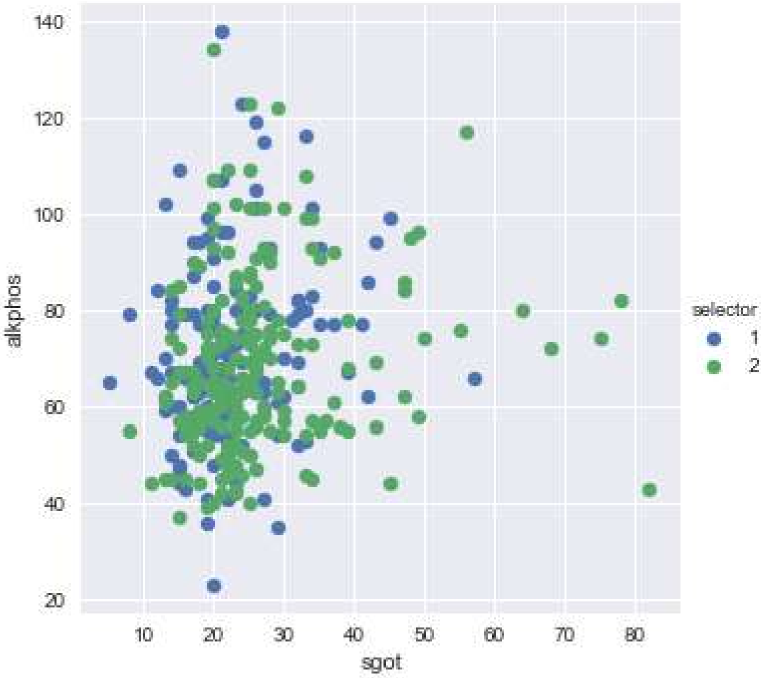
In [10]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "sgpt", "alkphos") \
   .add_legend();
plt.show();
```



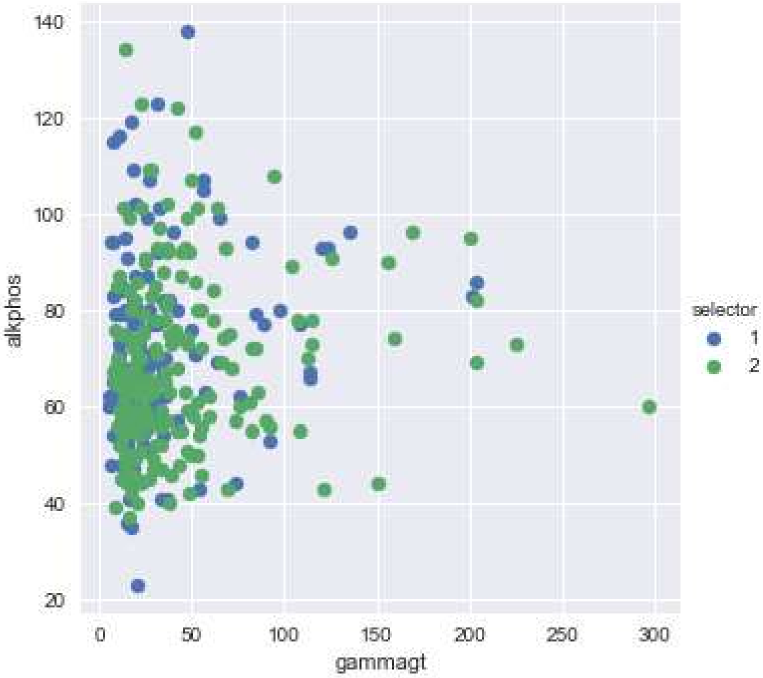
In [11]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "sgot", "alkphos") \
   .add_legend();
plt.show();
```



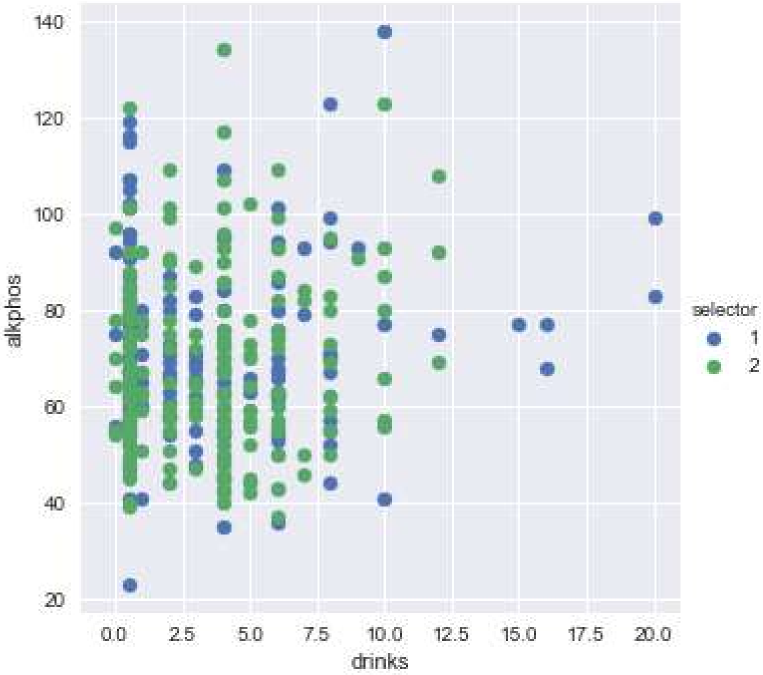
In [12]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "gammagt", "alkphos") \
   .add_legend();
plt.show();
```



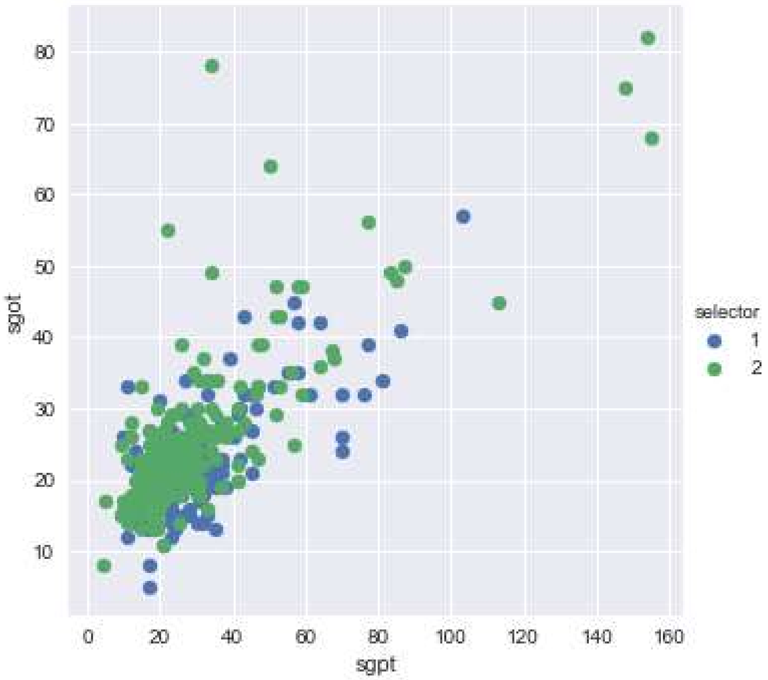
In [13]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "drinks", "alkphos") \
   .add_legend();
plt.show();
```



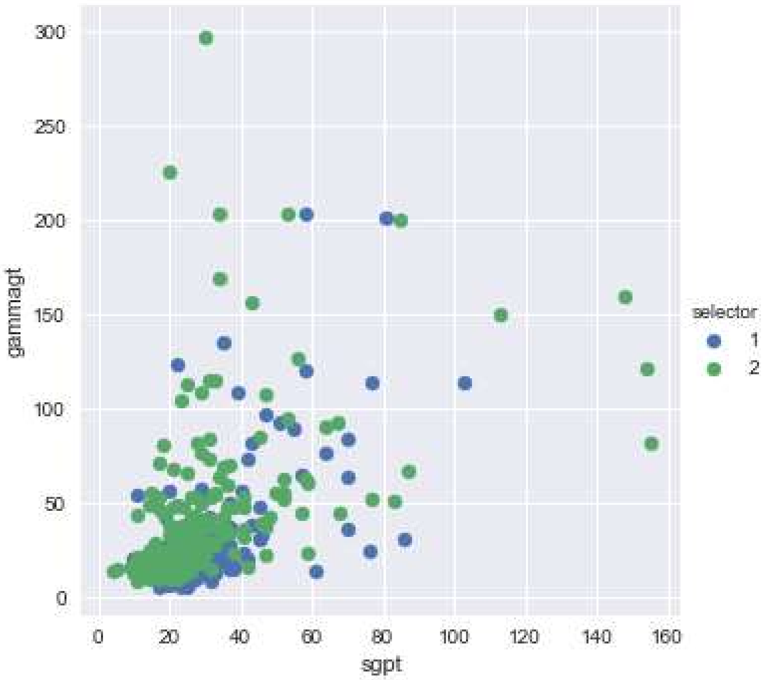
In [14]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "sgpt", "sgot") \
   .add_legend();
plt.show();
```



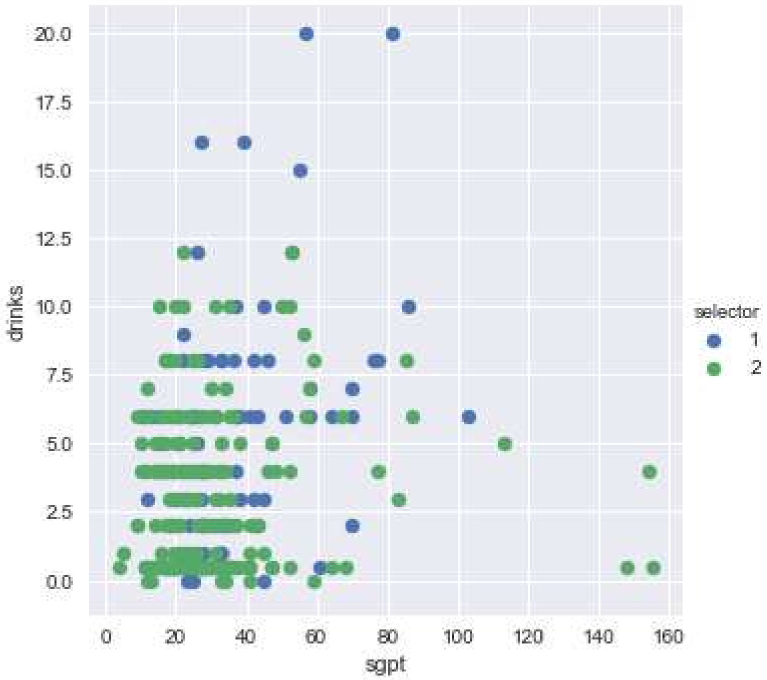
In [15]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "sgpt", "gammagt") \
   .add_legend();
plt.show();
```



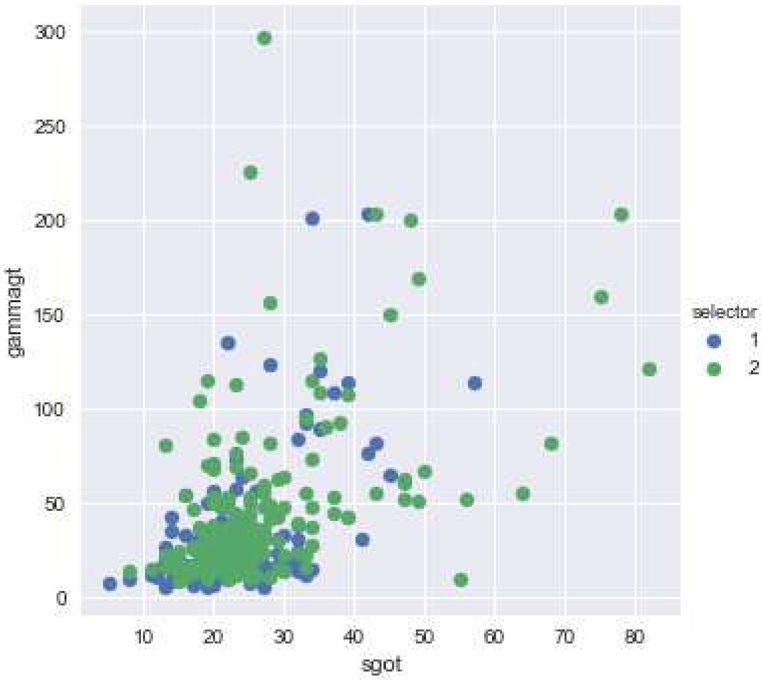
In [16]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "sgpt", "drinks") \
   .add_legend();
plt.show();
```



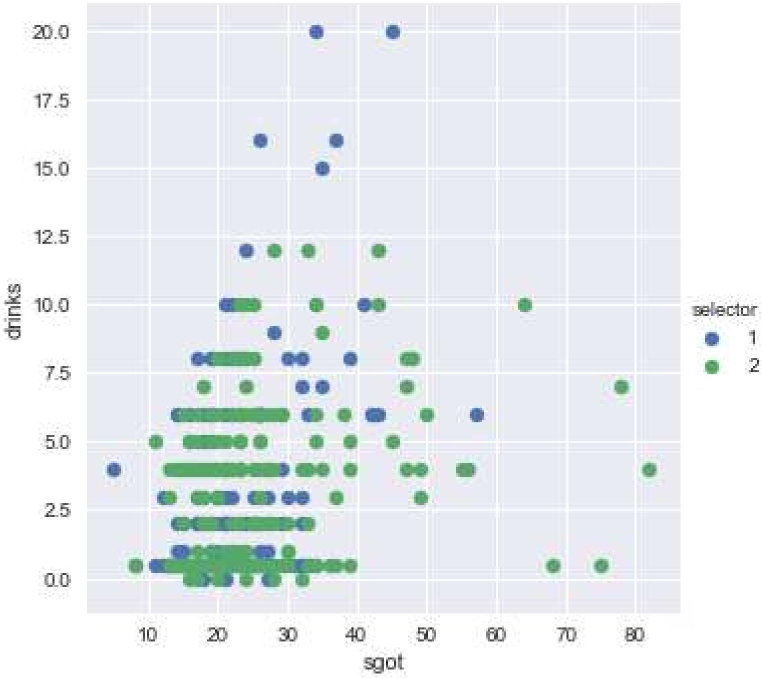
In [17]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "sgot", "gammagt") \
   .add_legend();
plt.show();
```



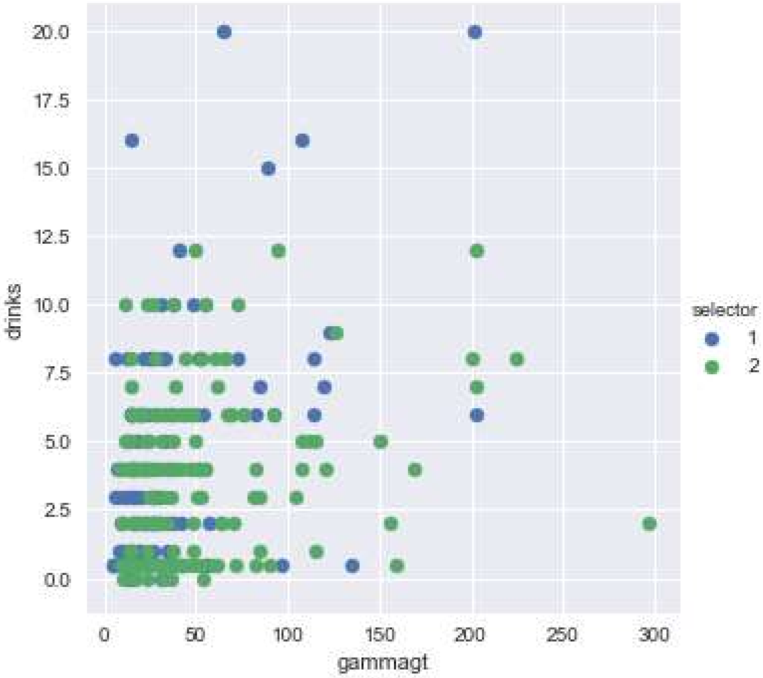
In [18]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "sgot", "drinks") \
   .add_legend();
plt.show();
```



In [19]:

```
sns.FacetGrid(bupa, hue="selector", size=5) \
   .map(plt.scatter, "gammagt", "drinks") \
   .add_legend();
plt.show();
```



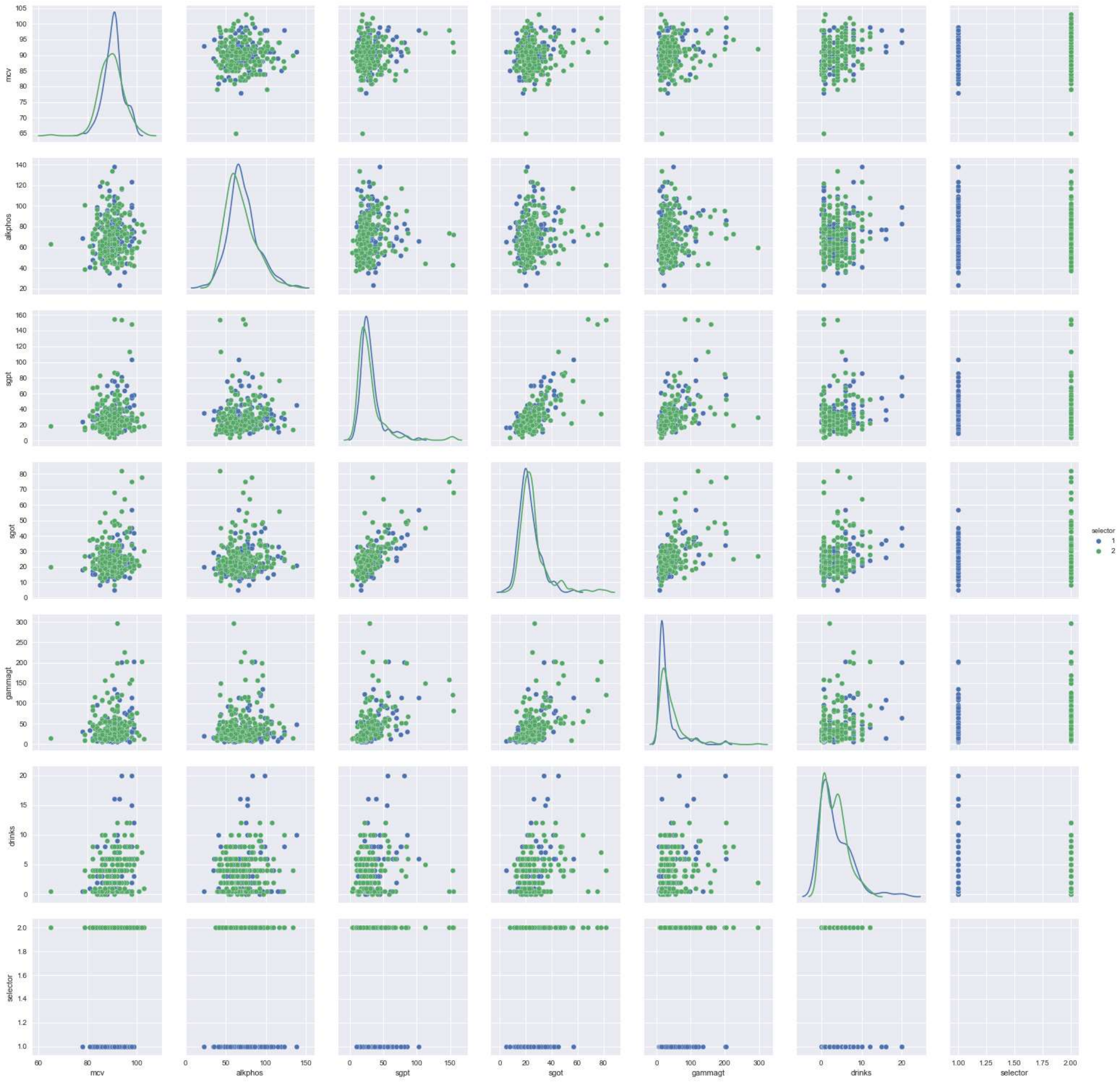
Pair-plot

In [20]:

```
# pairwise scatter plot: Pair-Plot
# Dis-advantages:
##Can be used when number of features are high.
##Cannot visualize higher dimensional patterns in 3-D and 4-D. Only possible to view 2D
 patterns.
plt.close();
sns.pairplot(bupa, hue="selector", size=3, diag_kind="kde");
plt.show()


# NOTE: the diagnol elements are PDFs for each feature. PDFs are expalined below.
```

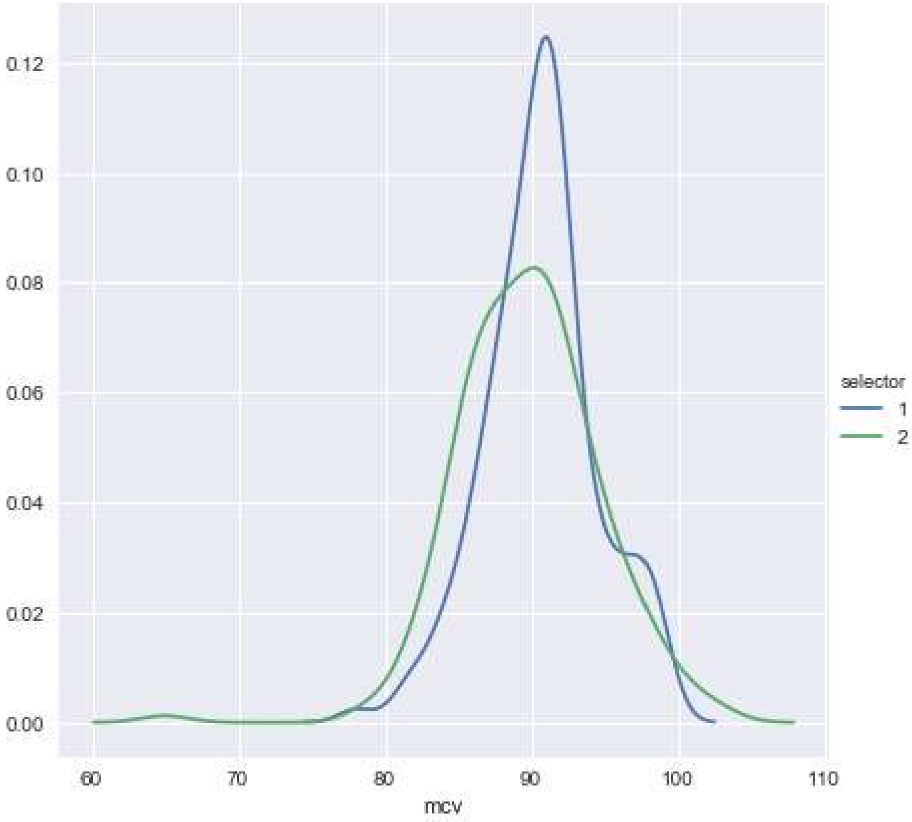
```
C:\Users\Admin\Anaconda2\lib\site-packages\statsmodels\nonparametric\kde.p
y:494: RuntimeWarning: invalid value encountered in divide
  binned = fast_linbin(X,a,b,gridsize)/(delta*nobs)
C:\Users\Admin\Anaconda2\lib\site-packages\statsmodels\nonparametric\kde.p
y:494: RuntimeWarning: invalid value encountered in true_divide
  binned = fast_linbin(X,a,b,gridsize)/(delta*nobs)
C:\Users\Admin\Anaconda2\lib\site-packages\statsmodels\nonparametric\kdeto
ols.py:34: RuntimeWarning: invalid value encountered in double_scalars
  FAC1 = 2*(np.pi*bw/RANGE)**2
```



PDF

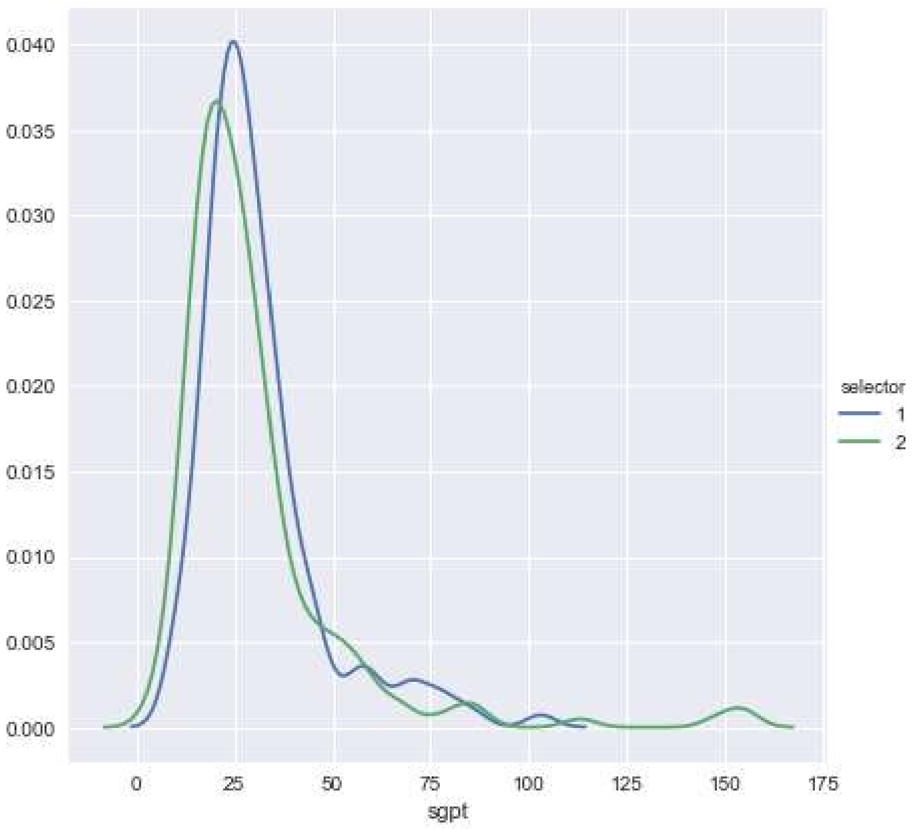
In [21]:

```
sns.FacetGrid(bupa, hue="selector", size=6) \
   .map(sns.kdeplot, "mcv") \
   .add_legend();
plt.show();
```



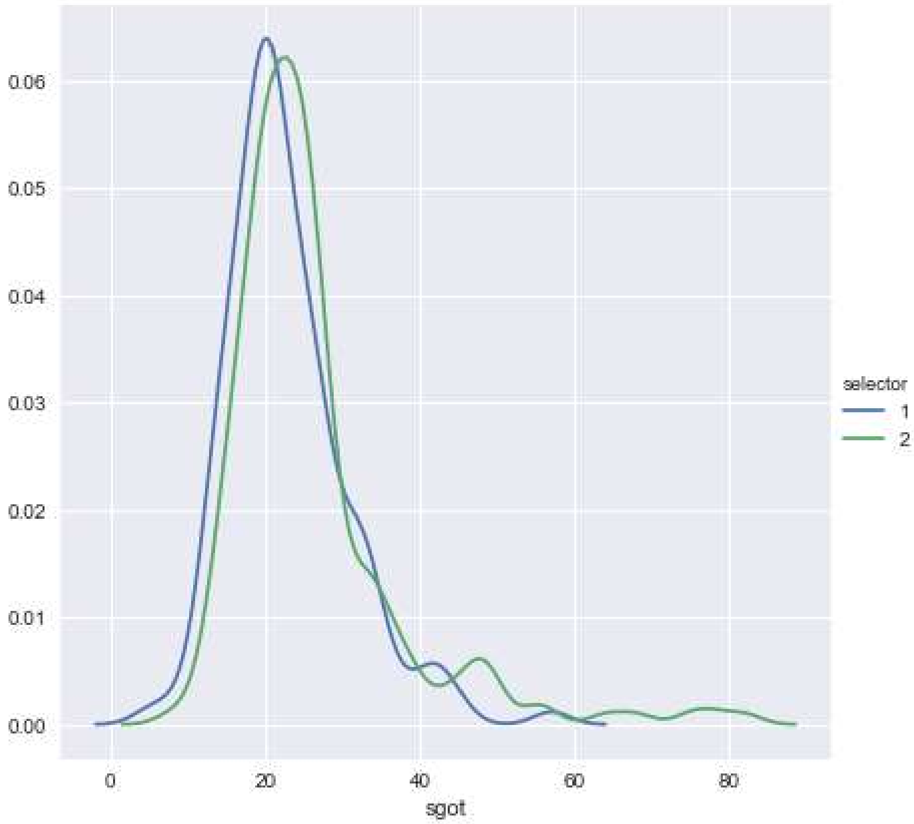
In [22]:

```
sns.FacetGrid(bupa, hue="selector", size=6) \
   .map(sns.kdeplot, "sgpt") \
   .add_legend();
plt.show();
```



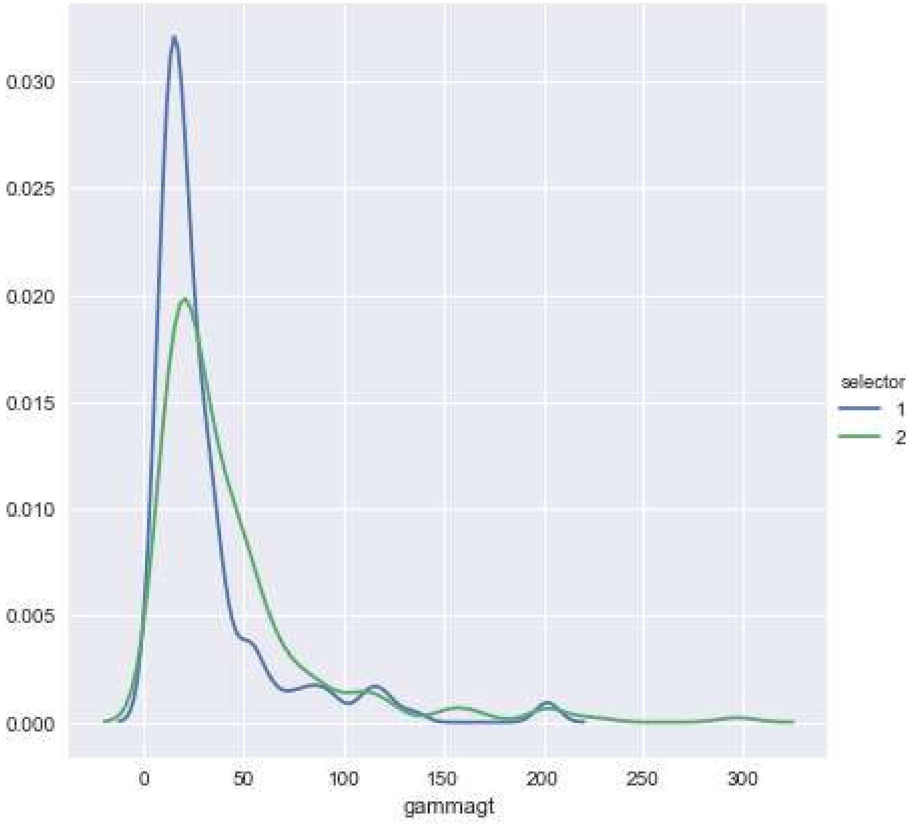
In [23]:

```
sns.FacetGrid(bupa, hue="selector", size=6) \
   .map(sns.kdeplot, "sgot") \
   .add_legend();
plt.show();
```



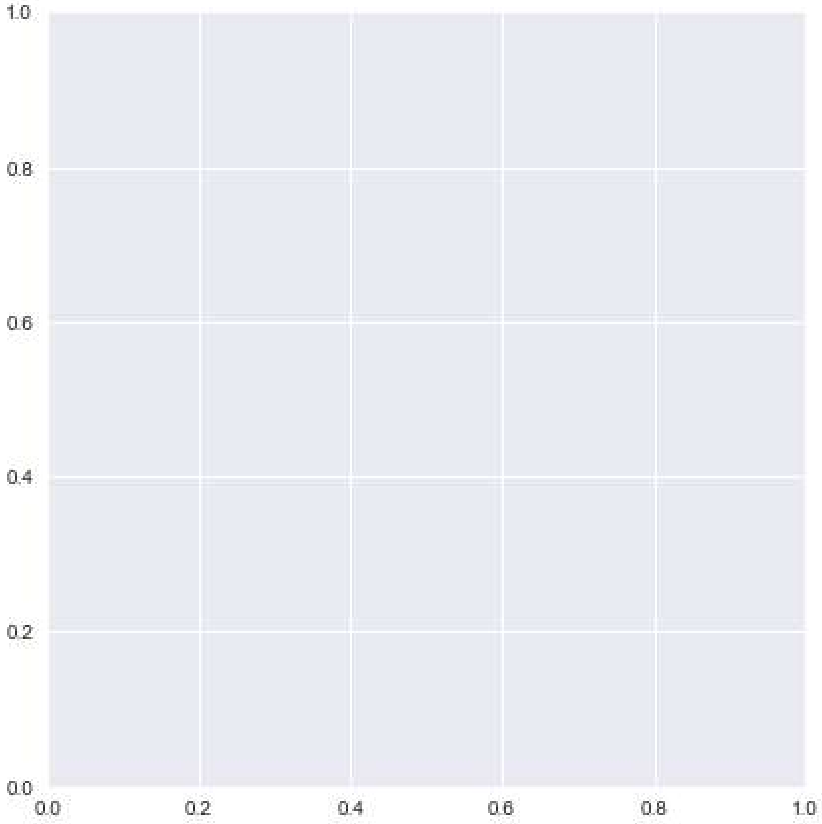
In [24]:

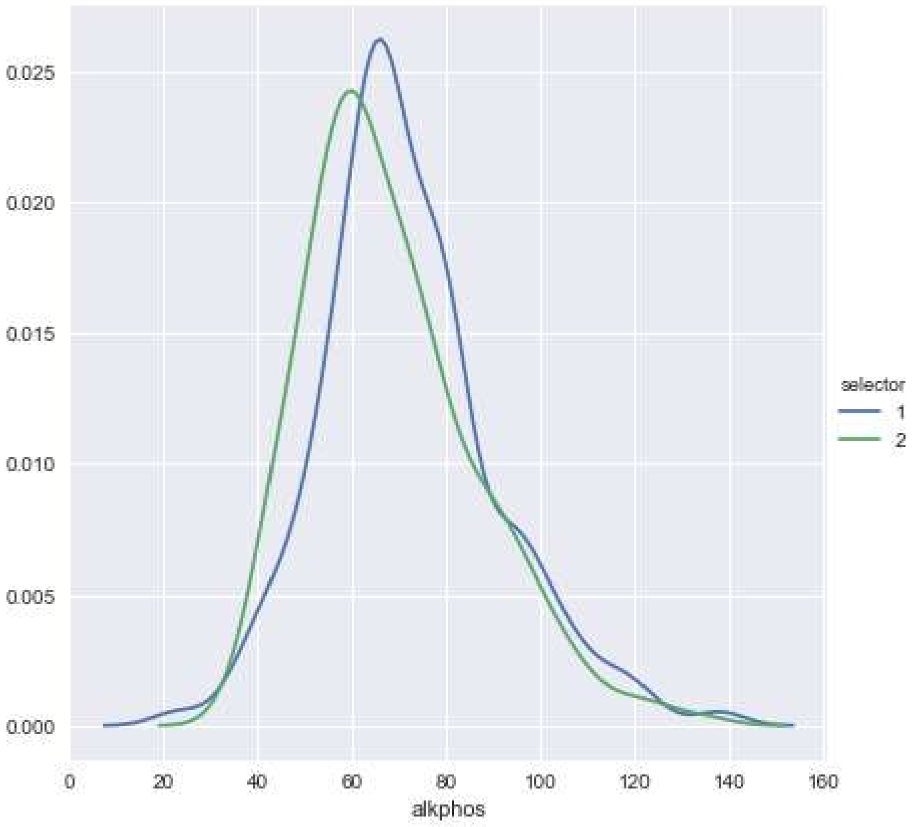
```
sns.FacetGrid(bupa, hue="selector", size=6) \
   .map(sns.kdeplot, "gammagt") \
   .add_legend();
plt.show();
```



In [26]:

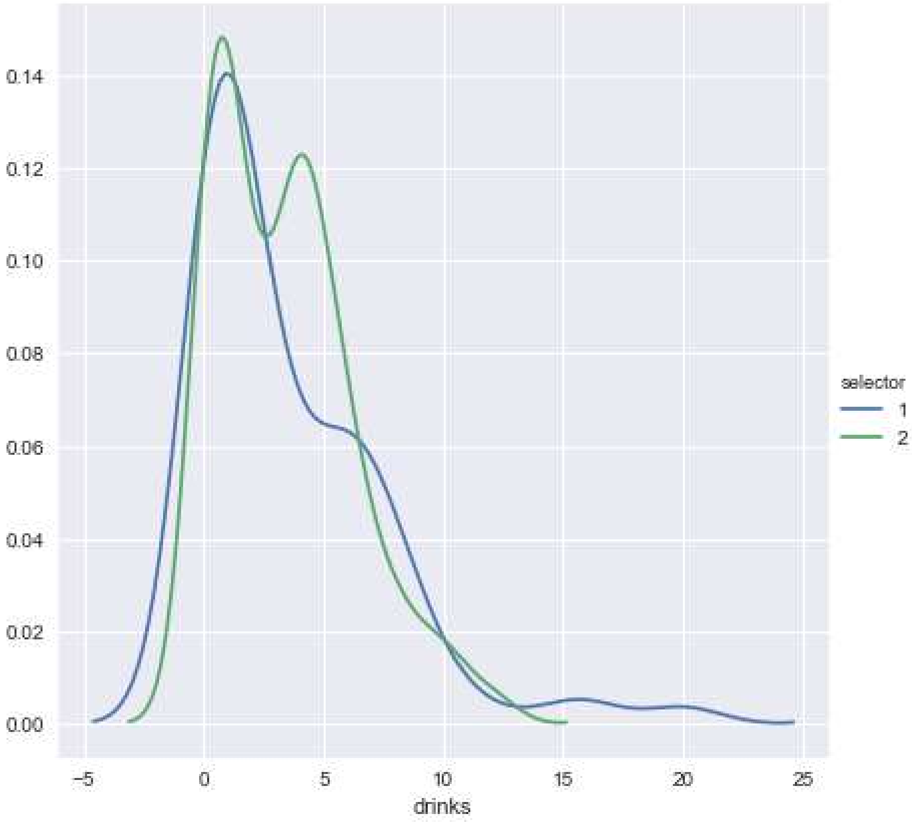
```
sns.FacetGrid(bupa, hue="selector", size=6) \
   .map(sns.kdeplot, "alkphos") \
   .add_legend();
plt.show();
```





In [27]:

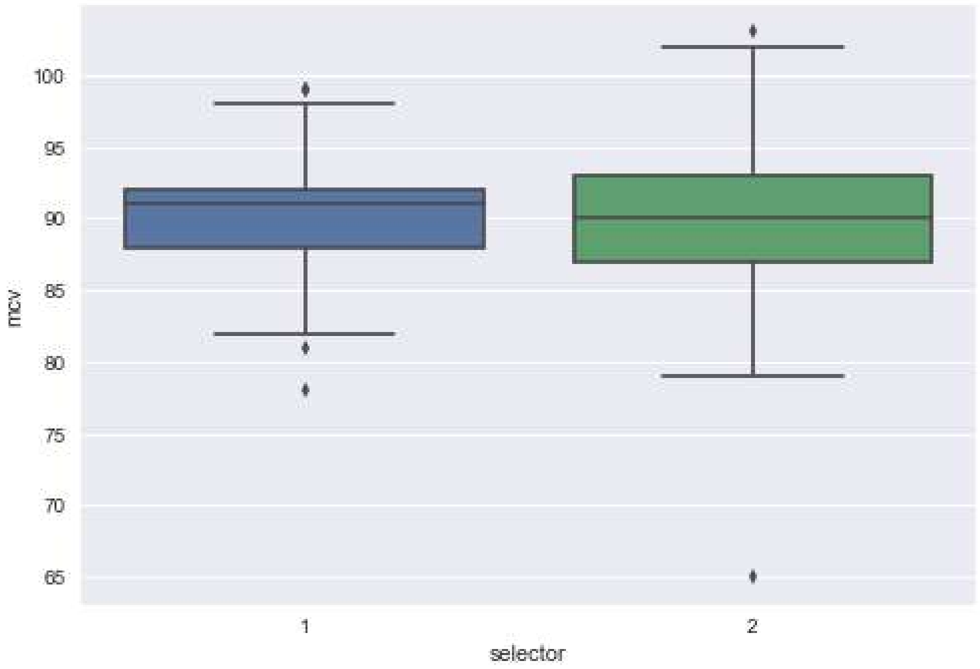
```
sns.FacetGrid(bupa, hue="selector", size=6) \
   .map(sns.kdeplot, "drinks") \
   .add_legend();
plt.show();
```



Box-plot

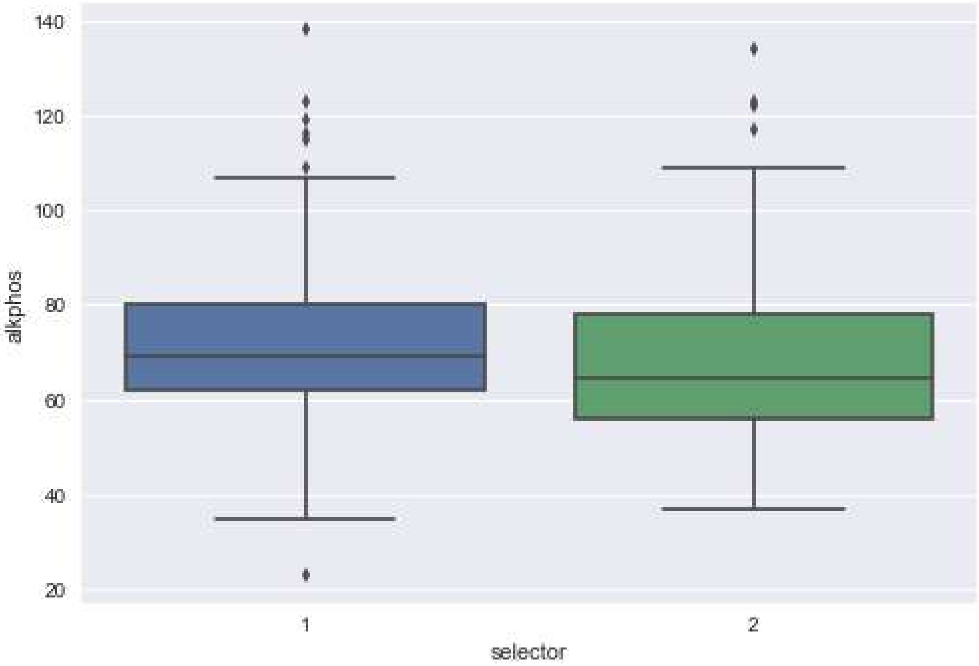
In [28]:

```
sns.boxplot(x='selector',y='mcv', data=bupa)
plt.show()
```



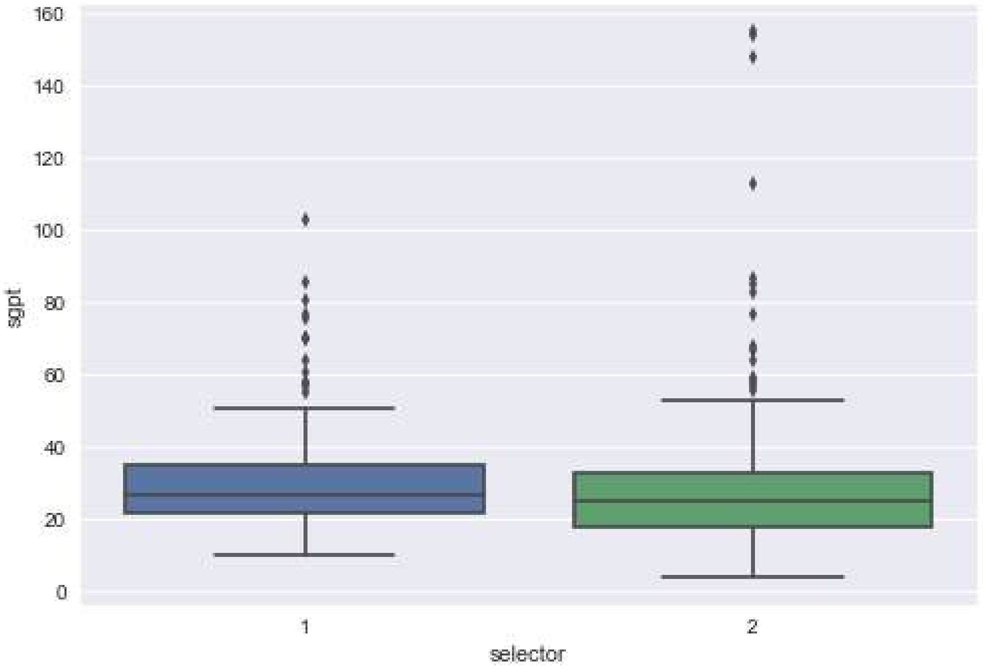
In [29]:

```
sns.boxplot(x='selector',y='alkphos', data=bupa)
plt.show()
```



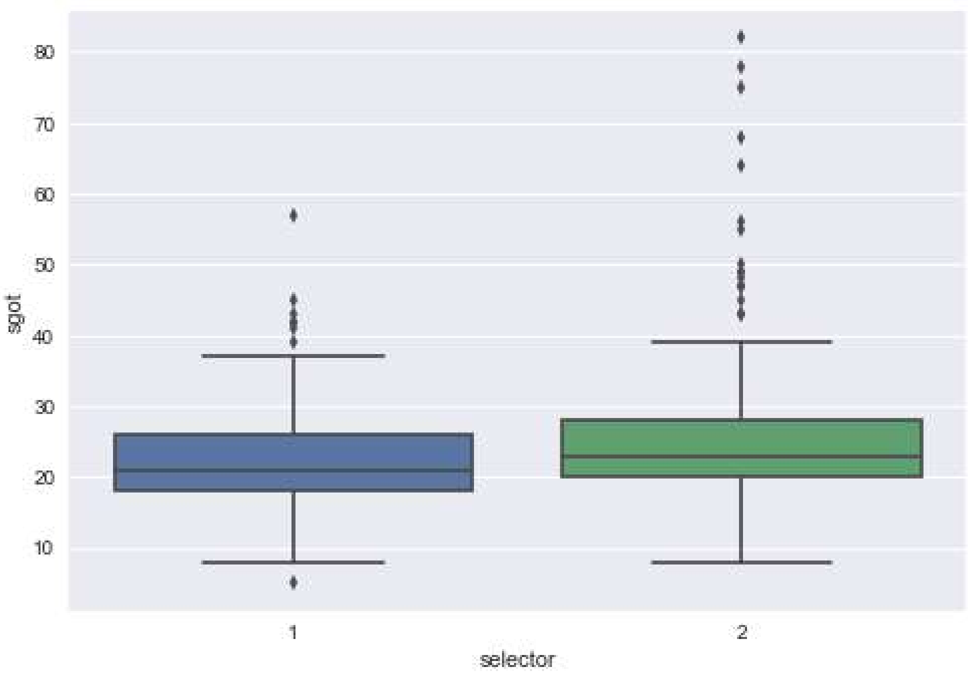
In [30]:

```
sns.boxplot(x='selector',y='sgpt', data=bupa)
plt.show()
```



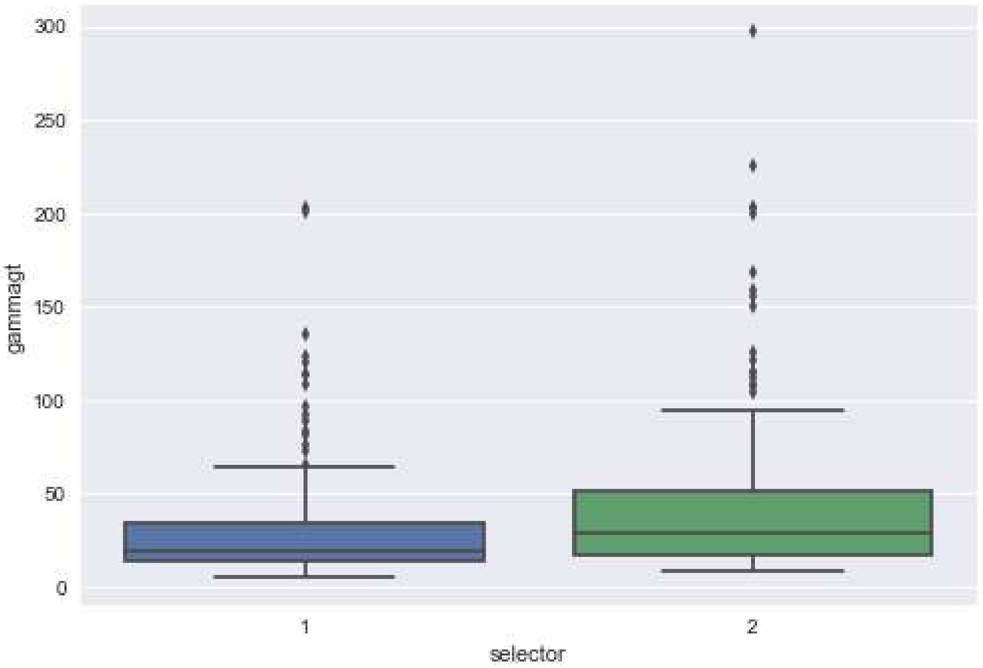
In [31]:

```
sns.boxplot(x='selector',y='sgot', data=bupa)
plt.show()
```



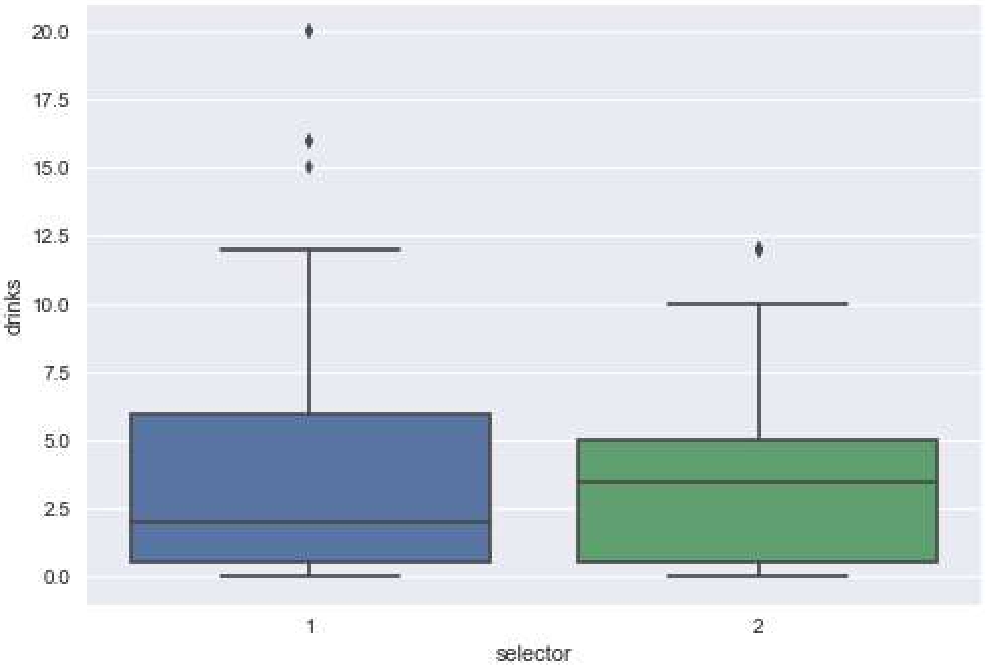
In [32]:

```
sns.boxplot(x='selector',y='gammagt', data=bupa)
plt.show()
```



In [33]:

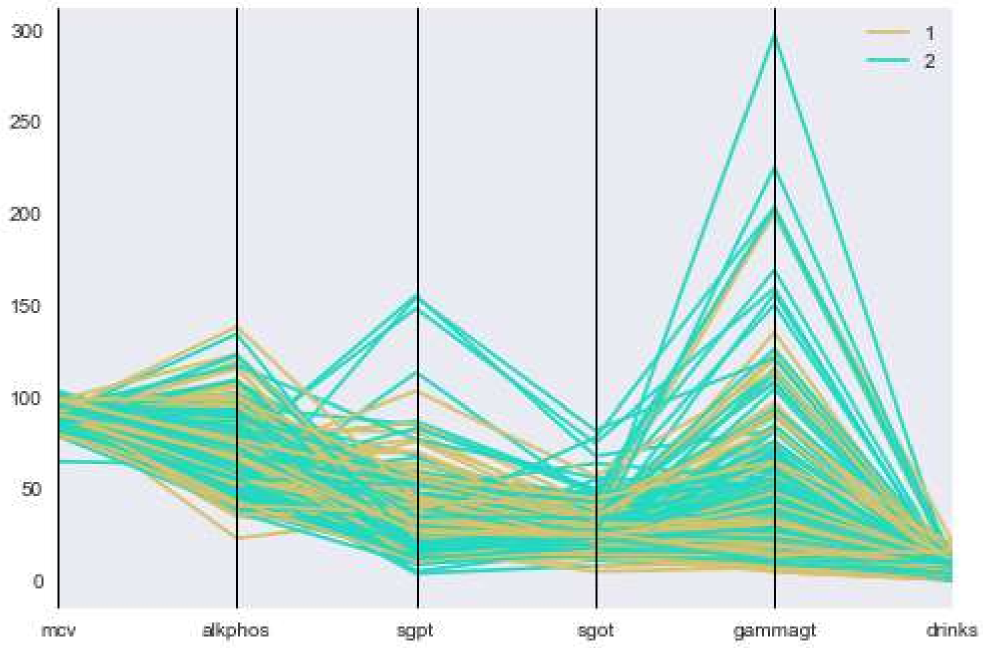
```
sns.boxplot(x='selector',y='drinks', data=bupa)
plt.show()
```



parallel coordinate

In [34]:

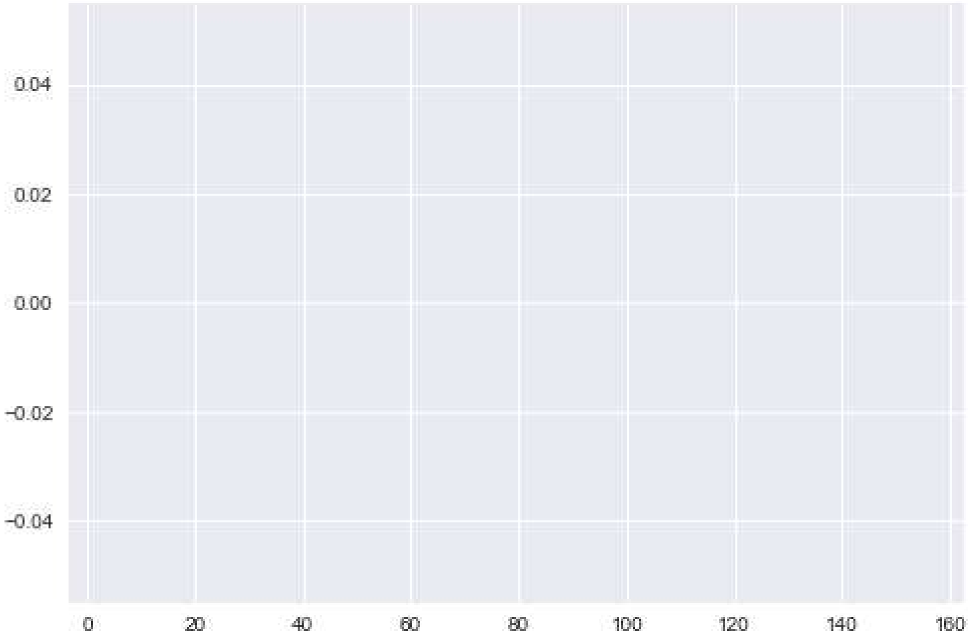
```
from pandas.plotting import parallel_coordinates
parallel_coordinates(bupa, "selector");
plt.show();
```



1D SCATTER PLOT

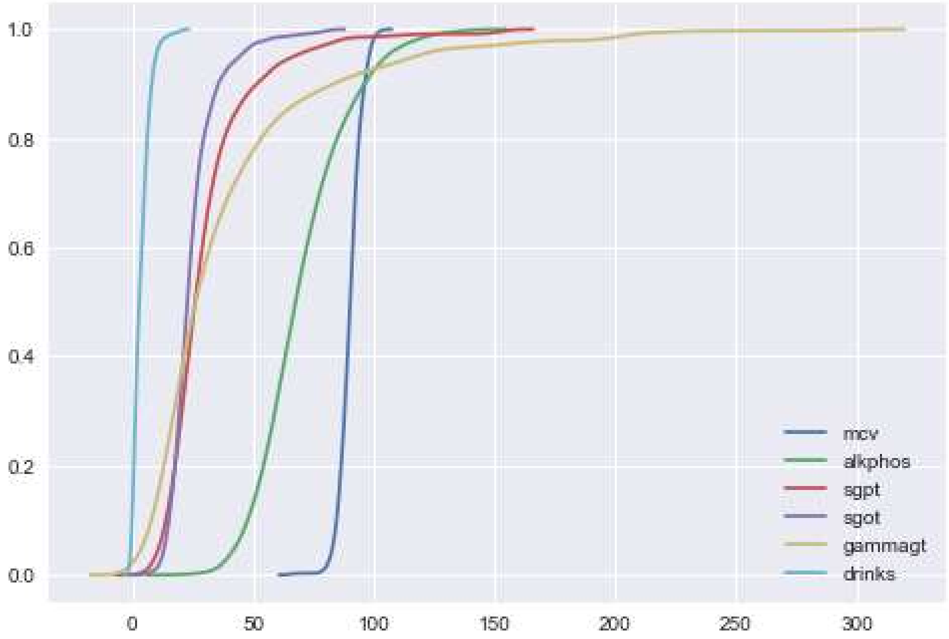
In [40]:

```
#1-D scatter plo of MCV
import numpy as np
plt.plot(bupa['sgpt'], np.zeros_like(bupa['sgpt']), 'x')
plt.show()
```



In [38]:

```
ax = sns.kdeplot(bupa['mcv'], cumulative=True)
ax = sns.kdeplot(bupa['alkphos'], cumulative=True)
ax = sns.kdeplot(bupa['sgpt'], cumulative=True)
ax = sns.kdeplot(bupa['sgot'], cumulative=True)
ax = sns.kdeplot(bupa['gammagt'], cumulative=True)
ax = sns.kdeplot(bupa['drinks'], cumulative=True)
plt.show()
```


mcv
alkphos
sgpt
sgot
gammagt
drinks