In [5]:
```
# COGS109
# Homework 3
# Gustav Sto. Tomas
# A15358078

import pandas as pd
import pandas.tseries
from pandas.core import datetools
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
from sklearn.metrics import confusion_matrix

df = pd.read_csv('hw3_divseq_data.csv')
```

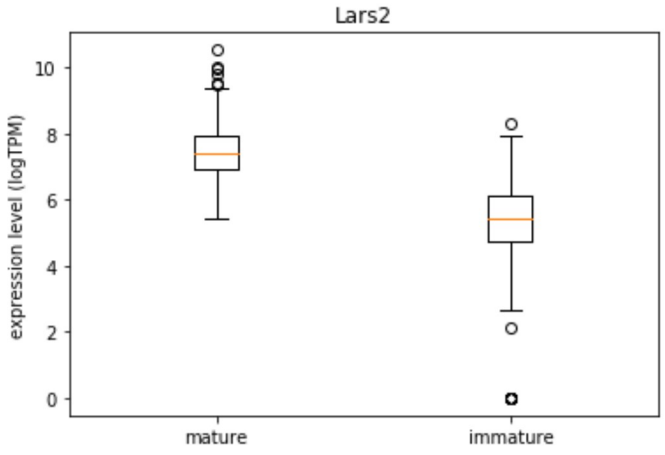
In [6]:
```
df.head()
```

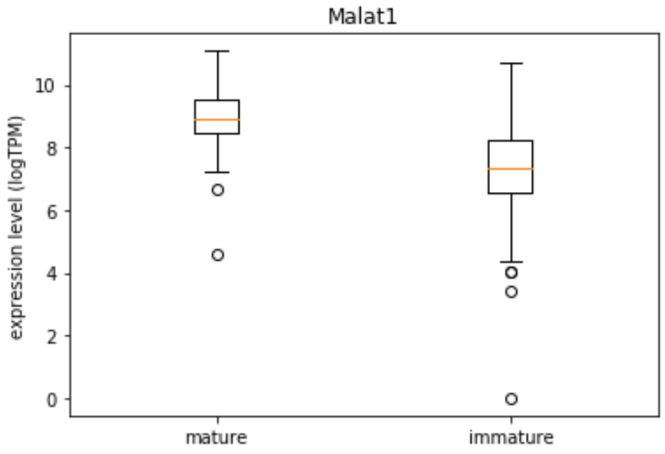
Out[6]:

	Lars2	Malat1	mature
0	9.95	6.69	1
1	10.54	8.53	1
2	6.58	8.74	1
3	7.49	9.09	1
4	7.42	9.87	1

In [7]:
```
lars = []
lars0 = []
for i in range(len(df.Lars2)):
    if df.mature[i] == 1:
        lars.append([df.Lars2[i]])
    else: lars0.append([df.Lars2[i]])



malat = []
malat0 = []
for i in range(len(df.Malat1)):
    if df.mature[i] == 1:
        malat.append([df.Malat1[i]])
    else: malat0.append([df.Malat1[i]])
```

In [8]:
```
#4a
plt.figure(1)
plt.boxplot([lars,lars0],labels = ['mature','immature'])
plt.ylabel('expression level (logTPM)')
plt.title('Lars2')

plt.figure(2)
plt.boxplot([malat,malat0],labels = ['mature','immature'])
plt.ylabel('expression level (logTPM)')
plt.title('Malat1')

plt.show()
```





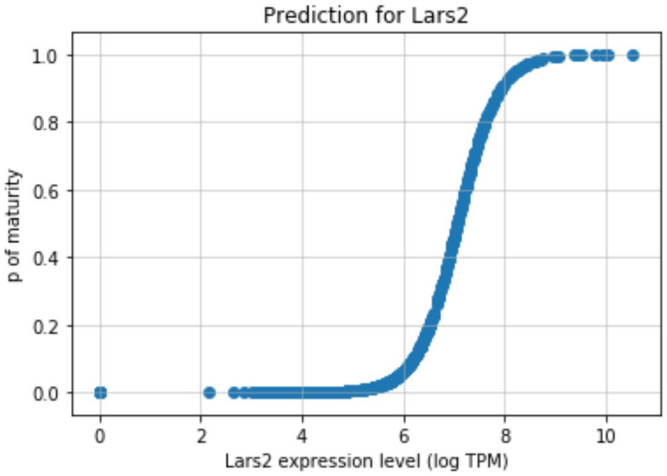
In [24]:
```
#4b)
print('Lars2 displays NO overlap in spread inside the interquartal range, so there
IS a difference between mature and immature neurons. We could almost use Lars2 to m
ake a perfect classifier. However, there are also a few extreme outliers of the imm
ature neurons that fall inside and above the median of the mature expression levels
, meaning that these would be falsely classified as mature, and therefore the class
ifier would not be perfect. Furthermore, in the plot for Malat1, the expression lev
els for mature show similar values as the immature ones in Lars2, which is why usin
g Lars2 as training set would not classify correctly for Malat1 as a test set. Conc
lusion: No.')
```

```
Lars2 displays NO overlap in spread inside the interquartal range, so there IS a
difference between mature and immature neurons. We could almost use Lars2 to mak
e a perfect classifier. However, there are also a few extreme outliers of the im
mature neurons that fall inside and above the median of the mature expression le
vels, meaning that these would be falsely classified as mature, and therefore th
e classifier would not be perfect. Furthermore, in the plot for Malat1, the expr
ession levels for mature show similar values as the immature ones in Lars2, whic
h is why using Lars2 as training set would not classify correctly for Malat1 as
a test set. Conclusion: No.
```

In [12]:
```
#4c)

logreg = smf.logit(formula = 'mature ~ Lars2', data=df).fit()
print(logreg.summary())
print(logreg.pvalues)
print('p-value for Lars2 coeffecient is 3.455778e-34; my conclusion is that there i
s a significantly positive correlation between Lars2 neurons\' expression values an
d maturity==1.')
```

```
Optimization terminated successfully.
         Current function value: 0.235975
         Iterations 9
                           Logit Regression Results
==============================================================================
Dep. Variable:                 mature   No. Observations:                  817
Model:                          Logit   Df Residuals:                      815
Method:                           MLE   Df Model:                            1
Date:                Sat, 21 Oct 2017   Pseudo R-squ.:                  0.5310
Time:                        14:30:06   Log-Likelihood:                -192.79
converged:                       True   LL-Null:                       -411.04
                                        LLR p-value:                6.284e-97
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -17.9775      1.432    -12.556      0.000     -20.784     -15.171
Lars2          2.5422      0.209     12.191      0.000       2.134       2.951
==============================================================================
Intercept    3.679364e-36
Lars2        3.455778e-34
dtype: float64
p-value for Lars2 coeffecient is 3.455778e-34; my conclusion is that there is a
significantly positive correlation between Lars2 neurons' expression values and
maturity==1.
```

In [14]:
```
#4d)

p = logreg.predict(df.Lars2)

fig, ax = plt.subplots()
ax.scatter(df.Lars2, p)
plt.xlabel('Lars2 expression level (log TPM)')
plt.ylabel('p of maturity')
plt.title('Prediction for Lars2')
ax.grid(linewidth=0.5)
plt.show()


p8 = []
for i in range(len(df.Lars2)):
    if df.Lars2[i] == 8.0:
        p8.append(p[i])
    if df.Lars2[i] == 7.9:
        p8.append(p[i])
    if df.Lars2[i] == 8.1:
        p8.append(p[i])
print(p8)

print('as there is no true value 8.0 in Lars2, our closest value is 8.1, for which
the predicted probability is 0.93: a very strong probability that the neuron is mat
ure.')
```




```
[0.93179368611821334, 0.93179368611821334]
as there is no true value 8.0 in Lars2, our closest value is 8.1, for which the
predicted probability is 0.9: a very strong probability that the neuron is matur
e.
```

In [15]:
```
#4e)

df['pred']=p
df['pred_Mature'] = 1*(df.pred > 0.5)
df.head(10)

correct = []
for i in range(len(df.pred_Mature)):
    if df.pred_Mature[i] == True:
        correct.append(df.pred_Mature[i])

print('correctly predicted mature neurons:', len(correct))
print('total amount of predictions:', len(df.pred))

sensitivity = len(correct)/len(df.pred)
print('sensitivity:', sensitivity)
```

```
correctly predicted mature neurons: 144
total amount of predictions: 817
sensitivity: 0.1762545899632803
```

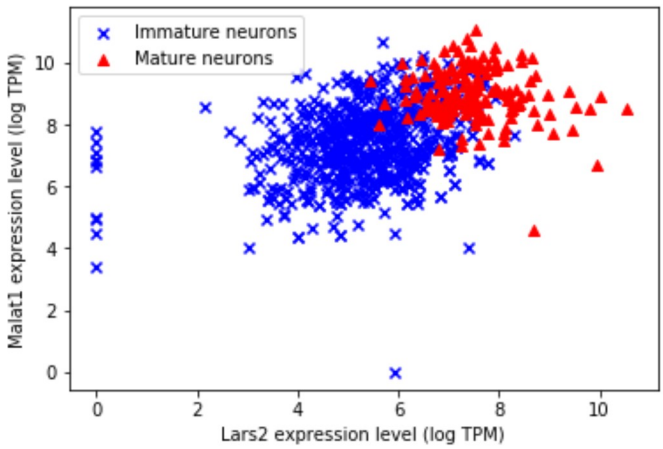
In [17]:
```
#4f)

df['pred_Immature'] = 1 *(df.pred < 0.5)

correct_im = []
for i in range(len(df.pred_Immature)):
    if df.pred_Immature[i] == True:
        correct_im.append(df.pred_Immature[i])

print('correctly predicted immature neurons:', len(correct_im))
print('total anount of predictions:', len(df.pred))
specificity = len(correct_im)/len(df.pred)
print('specificity:', specificity)
```

```
correctly predicted immature neurons: 673
total anount of predictions: 817
specificity: 0.8237454100367197
```

In [19]:
```
#4g

df['pred_not_immature'] = 1 *(df.pred > 0.2)

correct_nim = []
for i in range(len(df.pred_not_immature)):
    if df.pred_not_immature[i] == True:
        correct_nim.append(df.pred_not_immature[i])

correct_nim2 = []
for i in range(len(df.pred_not_immature)):
    if df.pred_not_immature[i] == True and df.pred_Immature[i] == True:
        correct_nim2.append(df.pred_not_immature[i])


print('correctly predicted \"not immature\" neurons:', len(correct_nim))
print('total amount of predictions:', len(df.pred))
sensitivity_nim = len(correct_nim)/len(df.pred)

specificity_nim = len(correct_nim2)/len(df.pred)
print('sensitivity:', sensitivity_nim)
print('specificity:', specificity_nim)

print('sensitivity increase because we get fewer false negatives. specificity decre
ase because we get more false positives. we may want to set the classification thre
shold to 20% for non-binary classifiers, in which case we would maybe want an \"alm
ost mature neuron\" class. furthermore, if the data does not contain any values abo
ve 0.5, we may still want to discriminate between the data points that are closer t
o 0.5 than the ones closer to 0.')
```

```
correctly predicted "not immature" neurons: 236
total amount of predictions: 817
sensitivity: 0.28886168910648713
specificity: 0.11260709914320685
sensitivity increase because we get fewer false negatives. specificity decrease
because we get more false positives. we may want to set the classification thres
hold to 20% for non-binary classifiers, in which case we would maybe want an "al
most mature neuron" class. furthermore, if the data does not contain any values
above 0.5, we may still want to discriminate between the data points that are cl
oser to 0.5 than the ones closer to 0.
```

In [20]:
```
#4h

fig, ax = plt.subplots()
untrues = [i for i,val in enumerate(df.mature) if val != True]
trues = [i for i, val in enumerate(df.mature) if val==True]
ax.scatter(df.Lars2[untrues],df.Malat1[untrues],color='b',marker='x',label='Immatur
e neurons')
ax.scatter(df.Lars2[trues], df.Malat1[trues],color='r', marker='^',label='Mature ne
urons')
plt.xlabel('Lars2 expression level (log TPM)')
plt.ylabel('Malat1 expression level (log TPM)')
plt.legend()
plt.show()
```



In [21]:
```
#4i)

logreg2 = smf.logit(formula = 'mature ~ Lars2 + Malat1', data=df).fit()
print(logreg2.summary())
print(logreg2.pvalues)
print(logreg2.tvalues)
print('p-values are < 0.01 for Lars2 and Malat (and intercept), and t-statistics (=
z) are > 2 for both Lars and Malat1 (but not the intercept)')
```

```
Optimization terminated successfully.
         Current function value: 0.196827
         Iterations 9
                           Logit Regression Results
==============================================================================
Dep. Variable:                 mature   No. Observations:                  817
Model:                          Logit   Df Residuals:                      814
Method:                           MLE   Df Model:                            2
Date:                Sat, 21 Oct 2017   Pseudo R-squ.:                  0.6088
Time:                        14:36:27   Log-Likelihood:                -160.81
converged:                       True   LL-Null:                       -411.04
                                        LLR p-value:                2.122e-109
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -25.5697      2.177    -11.743      0.000     -29.838     -21.302
Lars2          2.3119      0.223     10.354      0.000       1.874       2.750
Malat1         1.0836      0.156      6.941      0.000       0.778       1.390
==============================================================================
Intercept    7.691410e-32
Lars2        3.995510e-25
Malat1       3.885592e-12
dtype: float64
Intercept   -11.742782
Lars2        10.354432
Malat1        6.941281
dtype: float64
p-values are < 0.01 for Lars2 and Malat (and intercept), and t-statistics (=z) a
re > 2 for both Lars and Malat1 (but not the intercept)
```

In [23]:
```
#4j

pred_both = df[['Lars2','Malat1']]

df['pred_both'] = logreg2.predict(pred_both)
df['pred_both_mature'] = 1*(df.pred_both > 0.5)
df['pred_both_immature'] = 1 *(df.pred_both < 0.5)


correct_mb = []
for i in range(len(df.pred_both_mature)):
    if df.pred_both_mature[i] == 1:
        correct_mb.append(df.pred_both_mature[i])

correct_imb = []
for i in range(len(df.pred_both_mature)):
    if df.pred_both_immature[i] == 1:
        correct_imb.append(df.pred_both_immature[i])



print('correctly predicted mature neurons:', len(correct_mb))
print('total abount of predictions:', len(df.pred_both_mature))

sensitivity = len(correct_mb)/len(df.pred_both_mature)
print('sensitivity:', sensitivity)

specificity = len(correct_imb)/len(df.pred_both_mature)
print('specificity:', specificity)
print('both sensitivity and specifiity are similar to the values in 43 and 4f, thou
gh sensitivity is slightly larer and specificity slightly smaller')
```

```
correctly predicted mature neurons: 154
total abount of predictions: 817
sensitivity: 0.18849449204406366
specificity: 0.8115055079559363
both sensitivity and specifiity are similar to the values in 43 and 4f, though s
ensitivity is slightly larer and specificity slightly smaller
```