1. ISLR chapter 4, exercise 4 (page 168-169)

a)

if .05 < x <.95, we will have an interval[x.0.05,x+0.95], and a constant lenght of 0.1
if x<.05. we will observe the interval [0,x+0.05], and a changing lenght of (x+.05)

Also, if x>.95, we will observe the interval [x-.05, 1.0],and a changing lenght of (1.05-x)

Therefore, to compute the average fraction,we do intergal, we manipulate 100 for easiler computation of %

.9*0.1 + ∫0 to .05 (x+.05)dx+∫0.95 to 1 (1.05-x)dx == 9.75%

So,on average, the fraction of available observations we will use to make the prediction is 9.75%.

b)

Assumee X1 and X2 are independant var, then the fraction of available observations

we will use to make the prediction is 9.75%×9.75%=0.950625%.

c)

Known from a and b, the fraction of available observations we will use to make the prediction is 9.75 ^ 100 is closed to 0%.

d)

From a to c, we know that when p is a large value, (9.75%)^p is getting very closed to 0.

So formattly, we define when p→∞, we have limp→∞(9.75%)p=0.

e)

p=1, we have length=0.1

p=2, we have length=0.11/2

p=100, we have length=0.11/100.

2. ISLR chapter 4, exercise 6

a)

p̂ (X)= e ^(–6+0.05X1+X2) / (1 + e ^ (–6+0.05X1+X2)) = 0.3775.

so the prob for a student who studies for 40 hours and has an undergrad GPA of 3.5 gets an A in the class is 0.3775

b) So we set p̂ (X) to be 0.5

e ^ (–6+0.05X1+3.5) / (1 + e ^ (–6+0.05X1+X2)) = 0.5,

So, e−6+0.05X1+3.5=1.

Therefore, we get X1=50

3. ISLR chapter 4, exercise 7

when X = 4, by plugging value into the equation, we have

p1(4)= 0.8e−(1/72)(4−10)20.8e−(1/72)(4−10)2+0.2e−(1/72)(4−0)2=0.752;

so the probability that a company will issue a dividend this year given that its percentage return was X=4 last year is 0.752.

4 coding part

Most neurons in the brain develop before you are born and remain with you throughout your life. A small but important part of the brain called the dentate gyrus of the hippocampus continues to create new neurons past birth and into adulthood. These “adult newborn neurons” are thought to be important for creating distinct memories of similar events. In this problem, we will use a recently published data set containing gene expression measurements from single neurons to classify cells by their age. The study by Habib et al. is titled “ Div-Seq: Single-nucleus RNA-Seq reveals dynamics of rare adult newborn neurons ”

In [210]:

```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

#read data
df = pd.read_csv("/Users/xuzhaokai/Desktop/109 HW3 due 10:23/hw3_divseq_data.c
data = df.values # pd dataFrate to matrix

print data.shape
print data
```

executed in 32ms, finished 21:59:54 2018-10-23

```
(817, 3)
[[ 9.95  6.69  1.  ]
 [10.54  8.53  1.  ]
 [ 6.58  8.74  1.  ]
 ...
 [ 3.98  6.51  0.  ]
 [ 4.9   6.16  0.  ]
 [ 3.38  4.95  0.  ]]
```

In [6]:

```
Lars2 = data[:,0]
Malat1 = data[:,1]
mature = data[:,2]
```

executed in 6ms, finished 01:49:50 2018-10-23

a. Create a box plot showing the expression level of Lars2 for immature and mature neurons.

Do the same for Malat1 .

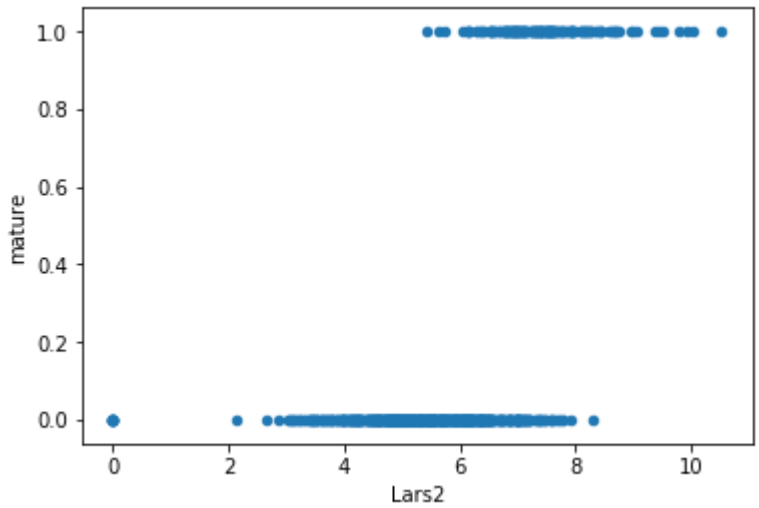
In [7]:

```
1 df.plot.scatter("Lars2","mature")
```

executed in 605ms, finished 01:49:59 2018-10-23

Out[7]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c19bb45d0>
```



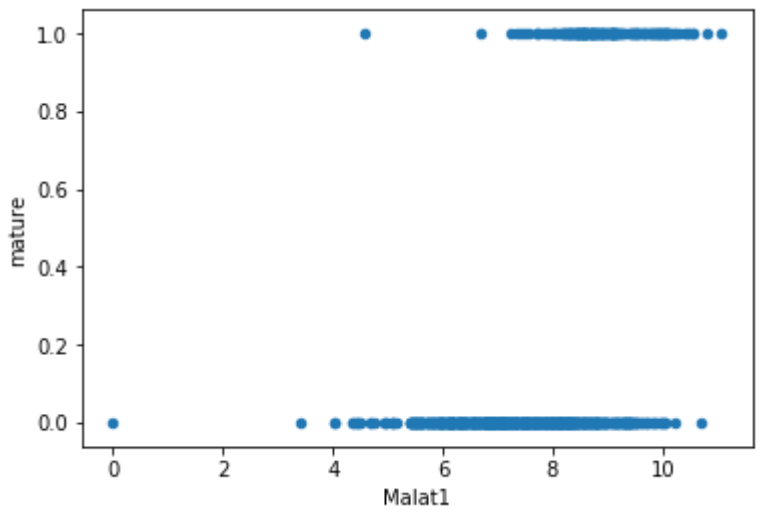
In [8]:

```
1 df.plot.scatter("Malat1","mature")
```

executed in 320ms, finished 01:50:01 2018-10-23

Out[8]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x106c1aa90>
```




```
1 #### b. Based on these plots, comment on whether you expect that
2 #### a classifier could perfectly predict a neuron's maturity based on Lars2 expression
  alone.
3
4 No because there are still a range of overlap between label 0 and label 1
5 for Lars2 from 5 to 8, the classifier can not prefectly predict what the label
  is
6 only based on Lars2
```

c. Fit a logistic regression to predict mature based on Lars2 alone; do not use Malat1 .

In [67]:

```
import numpy as np
import statsmodels.api as sm # statsmodels library
# import statsmodels.formula.api as smf
# same as import statsmodels.api as sm BUT no need to add const

#add const to gain intercept
Lars2_const = sm.add_constant(df["Lars2"])
print(Lars2_const.head())

model = sm.Logit(mature,Lars2_const)
results = model.fit()
print(results.summary2())
```

executed in 198ms, finished 10:14:25 2018-10-23

```
   const  Lars2
0    1.0   9.95
1    1.0  10.54
2    1.0   6.58
3    1.0   7.49
4    1.0   7.42
Optimization terminated successfully.
         Current function value: 0.235975
         Iterations 9
                         Results: Logit
=================================================================
Model:              Logit            No. Iterations:   9.0000
Dependent Variable: y                Pseudo R-squared: 0.531
Date:               2018-10-23 10:14 AIC:              389.5837
No. Observations:   817              BIC:              398.9950
Df Model:           1                Log-Likelihood:   -192.79
Df Residuals:       815              LL-Null:          -411.04
Converged:          1.0000           Scale:            1.0000
-----------------------------------------------------------------
```

In [68]:

```
print(results.pvalues)
```

executed in 11ms, finished 10:14:31 2018-10-23

```
const    3.679364e-36
Lars2    3.455778e-34
dtype: float64
```

What is the p-value for coefficient (slope) of Lars2 ?

3.455778e-34

What can you infer, i.e. what conclusion can you draw?

since the p - value(the prob that we would observe the data if H0 were true) is small, p<0.05, we can draw the conclustion that there is a very strong positive relation between Lars2 and mature

d. Using your model, calculate the predicted probability that each neuron is mature, i.e.

p = P(mature | Lars2) . Make a plot showing Lars2 on the x-axis vs. p on the y-axis.

The plot should have a sigmoid shape.

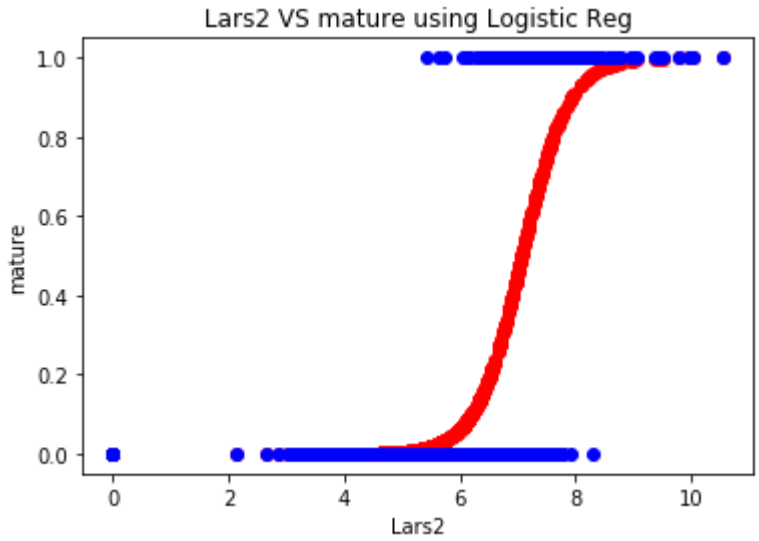
In [198]:

```
import numpy as np
import statsmodels.api as sm
import matplotlib.pyplot as plt

mature_predict = results.predict(Lars2_const)

plt.plot(Lars2, mature_predict, 'ro')
plt.plot(Lars2, mature, 'bo')
plt.xlabel("Lars2")
plt.ylabel("mature")
plt.title (" Lars2 VS mature using Logistic Reg ")
```

executed in 348ms, finished 21:49:18 2018-10-23

Out[198]:

```
<matplotlib.text.Text at 0x1c1e0c1390>
```



Based on this plot, what prediction would you make for the maturity of a cell with Lars2 = 8?

when Lars2 = 8, we predict the mature to be 1, which means true

e. Use a Bayesian classification criterion to predict, for each cell, whether or not it is mature.

In [199]:

```
# Recall that a Bayesian classifier chooses the most likely category; in this
# should predict “mature” whenever . Using these predictions, P(mature | Lars2
# sensitivity of your classifier, i.e. the fraction of mature cells that are c
# mature.

predict_label_list = []

for item in mature_predict:
    if item < 0.5:
        predict_label_list.append(0)
    else:
        predict_label_list.append(1)

mature_counter=0
mature_size = 0
size = len(predict_label_list)
for i in range(size):
    if mature[i] == 1:
        mature_size+=1
    if predict_label_list[i] == mature[i] and  mature[i] ==1:
        mature_counter += 1

print "mature_size: ", mature_size
print "mature that are correctly predicted ", mature_counter
print " the fraction of mature cells that are correctly classified as mature i
print float(mature_counter)/float(mature_size) *100,"%"
```

executed in 33ms, finished 21:49:20 2018-10-23

```
mature_size:  165
mature that are correctly predicted  108
 the fraction of mature cells that are correctly classified as mature
is
65.4545454545 %
```

f. Compute the specificity of your classifier,

In [200]:

```
#i.e. the fraction of immature cells that are correctly classified as immature

immature_counter=0
immature_size = 0
size = len(predict_label_list)
for i in range(size):
    if mature[i] == 0:
        immature_size+=1
    if predict_label_list[i] == mature[i] and  mature[i] ==0:
        immature_counter += 1

print "immature_size: ", immature_size
print "immature that are correctly predicted ", immature_counter

print " the fraction of immature cells that are correctly classified as immatu
print float(immature_counter)/float(immature_size) *100,"%"


```

executed in 24ms, finished 21:49:23 2018-10-23

```
immature_size:  652
immature that are correctly predicted  616
 the fraction of immature cells that are correctly classified as immat
ure is
94.4785276074 %
```

g Try predicting the maturity level for each cell with a threshold of 20%,

In [201]:

```
# i.e. predict mature whenever P(mature | Lars2) > 0.2 . What are the sensitiv

predict_label_list = []

for item in mature_predict:
    if item < 0.2:
        predict_label_list.append(0)
    else:
        predict_label_list.append(1)

mature_counter=0
mature_size = 0
size = len(predict_label_list)
for i in range(size):
    if mature[i] == 1:
        mature_size+=1
    if predict_label_list[i] == mature[i] and  mature[i] ==1:
        mature_counter += 1

print "mature_size: ", mature_size
print "mature that are correctly predicted ", mature_counter
print " sensitivity == the fraction of mature cells that are correctly classif
print float(mature_counter)/float(mature_size) *100,"%"

immature_counter=0
immature_size = 0
size = len(predict_label_list)
for i in range(size):
    if mature[i] == 0:
        immature_size+=1
    if predict_label_list[i] == mature[i] and  mature[i] ==0:
        immature_counter += 1

print "\n\nimmature_size: ", immature_size
print "immature that are correctly predicted ", immature_counter

print " specificity == the fraction of immature cells that are correctly class
print float(immature_counter)/float(immature_size) *100,"%"
```

executed in 59ms, finished 21:49:25 2018-10-23

```
mature_size:  165
mature that are correctly predicted  150
 sensitivity == the fraction of mature cells that are correctly classi
fied as mature is
90.9090909091 %


immature_size:  652
immature that are correctly predicted  566
 specificity == the fraction of immature cells that are correctly clas
sified as immature is
86.8098159509 %
```

g. Explain why the sensitivity is increased, while the specificity is decreased.

By observing the previously printed diagram. The sensitivity is increased, while the specificity is decreased
is mainly a result of the distribution of the data.

```
if we set the threshold to be 0.5,
then a huge number of the mature data are misclassified to be immature,

while the immature data are mostly predicted correctly.
This results in a low sensitivity and a high specificity


And if we set the threshold to be 0.2
then the theshold is moving a bit left, so that less mature data are misclassified,

while a little bit more immature data  are misclassified.
This resulte in a much higher sensitivity and a bit lower specificity


From this experiment, we notice that there is a trade-off while selecting the threshold
Neither too large nor too small will not be good for the overall prediction.

#### In what circumstance might you prefer to use this classification threshold (20%)
#### instead of the Bayesian threshold (50%)?

When the distribution of matual data is mainly located on the right of threshold 20%
and the distribution of immatual data is mainly located on the left of the threshold 20%,
we prefer to have a threshold(20%) that will leads to a higher sensitivity and specificity

```

h. Now we will incorporate data from both genes to try to improve our prediction.

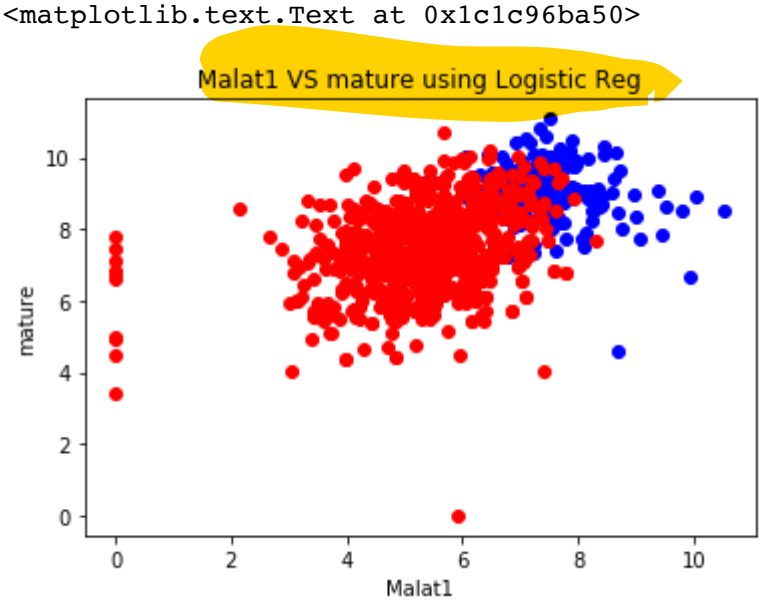
In [223]:

```
#  First, make a scatter plot showing Lars2 expression (x-axis) vs. Malat1 exp
# Use a different color and/or plot symbol for cells that are immature and mat
# Make sure to label the axes of the plot and include a legend explaining
# which color/symbol corresponds to which condition.

import matplotlib.pyplot as plt
import numpy as np

mature_list = []
immature_list = []

for item in data:
    if item[2] == 1:
        mature_list.append(item)
    else:
        immature_list.append(item)

mature_list = np.array(mature_list)
immature_list = np.array(immature_list)


plt.plot(mature_list[:,0], mature_list[:,1], 'o', color='blue');
plt.plot(immature_list[:,0], immature_list[:,1], 'o', color='red');
plt.xlabel("Malat1")
plt.ylabel("mature")
plt.
plt.title (" Malat1 VS mature using Logistic Reg ")

```

executed in 323ms, finished 22:15:25 2018-10-23

Out[223]:

```
<matplotlib.text.Text at 0x1c1c96ba50>
```



In [203]:

```
# i. Fit a logistic regression using both Lars2 and Malat1 as predictors. Prin
# summary table showing the coefficients, SE, t-statistic and p-value for each
import numpy as np
import statsmodels.api as sm # statsmodels library

#add const to gain intercept
newdata = data[:,0:2]
newdata = sm.add_constant(newdata)

model2 = sm.Logit(mature,newdata)
result2 = model2.fit()
print ("Notice that here x1 represents Lars2, and x2 represents Malat1, const
print(result2.summary2())
```

executed in 64ms, finished 21:50:28 2018-10-23

```
Optimization terminated successfully.
         Current function value: 0.196827
         Iterations 9
Notice that here x1 represents Lars2, and x2 represents Malat1, const
is the intercept
                         Results: Logit
=================================================================
Model:              Logit            No. Iterations:   9.0000
Dependent Variable: y                Pseudo R-squared: 0.609
Date:               2018-10-23 21:50 AIC:              327.6150
No. Observations:   817              BIC:              341.7320
Df Model:           2                Log-Likelihood:   -160.81
Df Residuals:       814              LL-Null:          -411.04
Converged:          1.0000           Scale:            1.0000
-----------------------------------------------------------------
         Coef.    Std.Err.      z      P>|z|    [0.025    0.975]
-----------------------------------------------------------------
const   -25.5697    2.1775  -11.7428  0.0000  -29.8375  -21.3019
x1        2.3119    0.2233   10.3544  0.0000    1.8743    2.7495
x2        1.0836    0.1561    6.9413  0.0000    0.7776    1.3895
=================================================================
```

Which predictors have a significant effect?

Apparently Lars2 has a more significant effect because its coef is larger than the coef of Malat1.

j. Use your new model to predict whether each neuron is mature, using a Bayesian decision threshold,

In [204]:

```
# i.e. P(mature | Malat1, Lars2) > 0.5 . What are the sensitivity and specific
# new prediction?


new_mature_predict = result2.predict(newdata)

new_predict_label_list = []

for item in new_mature_predict:
    if item < 0.5:
        new_predict_label_list.append(0)
    else:
        new_predict_label_list.append(1)

mature_counter=0
mature_size = 0
size = len(new_predict_label_list)
for i in range(size):
    if mature[i] == 1:
        mature_size+=1
    if new_predict_label_list[i] == mature[i] and  mature[i] ==1:
        mature_counter += 1

print "mature_size: ", mature_size
print "mature that is correctly predicted ", mature_counter
print " sensitivity == the fraction of mature cells that are correctly classif
print float(mature_counter)/float(mature_size) *100,"%"

immature_counter=0
immature_size = 0
size = len(new_predict_label_list)
for i in range(size):
    if mature[i] == 0:
        immature_size+=1
    if new_predict_label_list[i] == mature[i] and mature[i] ==0:
        immature_counter += 1

print "\n\nimmature_size: ", immature_size
print "immature that is correctly predicted ", immature_counter

print " specificity == the fraction of immature cells that are correctly class
print float(immature_counter)/float(immature_size) *100,"%"

```

executed in 60ms, finished 21:50:31 2018-10-23

```
mature_size:  165
mature that is correctly predicted  120
 sensitivity == the fraction of mature cells that are correctly classi
fied as mature is
72.7272727273 %


immature_size:  652
immature that is correctly predicted  618
 specificity == the fraction of immature cells that are correctly clas
sified as immature is
94.7852760736 %
```

Compare these values to the sensitivity and specificity you calculated in part (e)

In part e, we obtain the sensitivity == 65.4545454545 % and specificity == 94.4785276074 %

Now with two factors to predict, we get sensitivity == 72.7272727273 % and specificity == 94.7852760736 %

Happily, we improve the specificity slightly and we make a huge improvement on sensitivity. So we can conclude that in most case, build a model with more than one factor to predict should give a better prediction.

In [ ]:

```
1
```