In [1]:

```
import numpy as np
import numpy.linalg as la
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:

```
labels = np.genfromtxt('labels.txt',dtype='str',usecols=(0,), delimiter=',')
labelsid = {}
for i, label in enumerate(labels):
    labelsid[label] = i
subset_labels = ["radius (mean)", "perimeter (mean)","area (mean)", "symmetry (m
ean)"]
```

In [3]:

```
BM = {'B': -1, 'M': 1}
tumor_data = np.loadtxt("breast-cancer-train.dat", delimiter=",",
                        converters={1: lambda s: BM[s.decode('utf-8')]})
```

```
validate_data = np.loadtxt("breast-cancer-validate.dat", delimiter=",",
                           converters={1: lambda s: BM[s.decode('utf-8')]})
```

In [4]:

```
print(tumor_data.shape)
print(validate_data.shape)
```

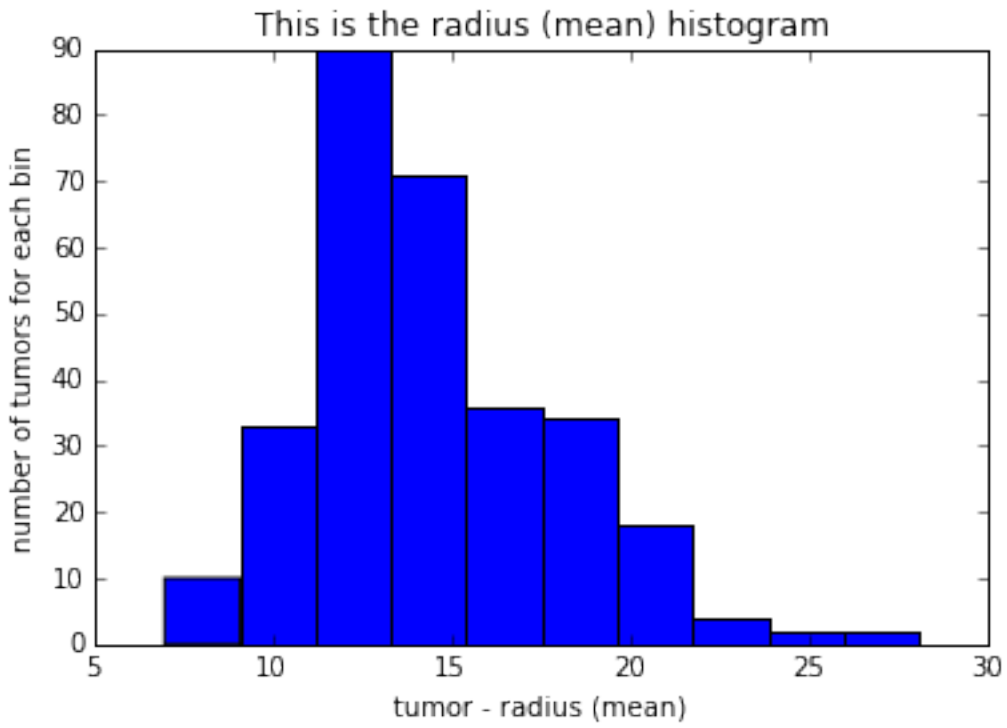
```
(300, 32)
(260, 32)
```

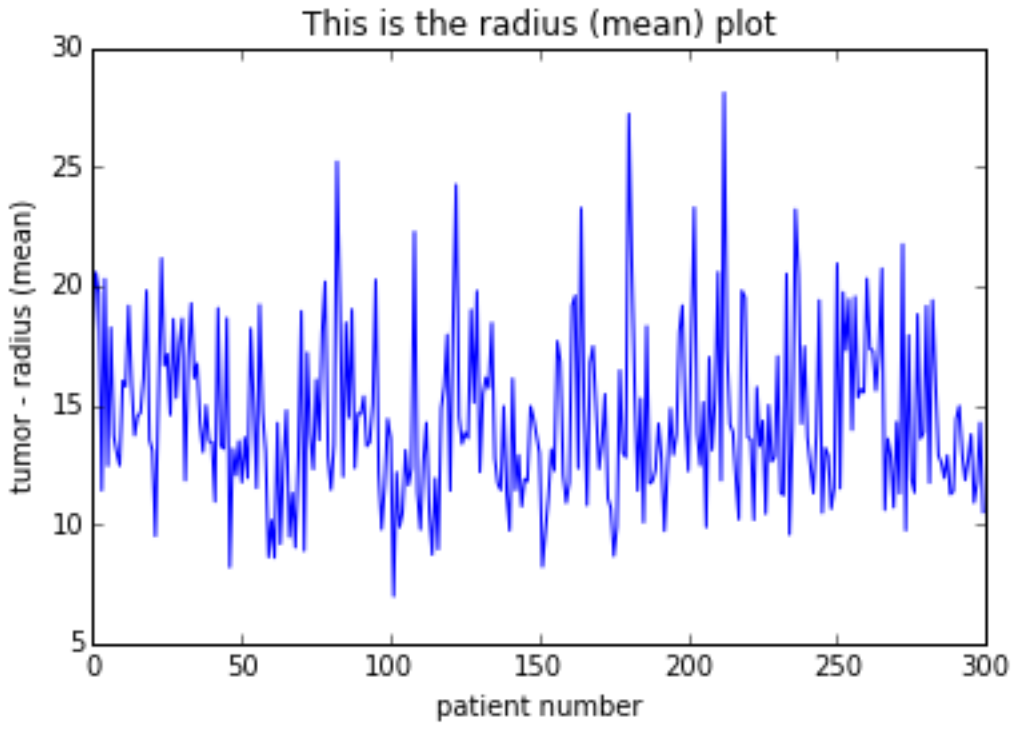
In [5]:

```
plt.figure(0)
plt.hist(tumor_data[:,labelsid["radius (mean)"]])
plt.title("This is the radius (mean) histogram")
plt.xlabel("tumor - radius (mean)")
plt.ylabel("number of tumors for each bin")
plt.figure(1)
plt.plot(tumor_data[:,labelsid["radius (mean)"]])
plt.title("This is the radius (mean) plot")
plt.xlabel("patient number")
plt.ylabel("tumor - radius (mean)")
```

Out[5]:

<matplotlib.text.Text at 0x104269908>





In [6]:

```
#  Construct the RHS
b = tumor_data[:,labelsid["Malignant/Benign"]]
v = validate_data[:,labelsid["Malignant/Benign"]]
```

In [7]:

```
#  Construct the Linear Models
#+ A: is the test data
#+ B: is the validation data
A_linear = tumor_data[:, 2:]
B_linear = validate_data[:, 2:]
```

In [8]:

```
#  Solve the Linear Model
Q, R = la.qr(A_linear, "complete")
weights_linear = la.solve(R[:30], Q.T.dot(b)[:30])
```

In [9]:

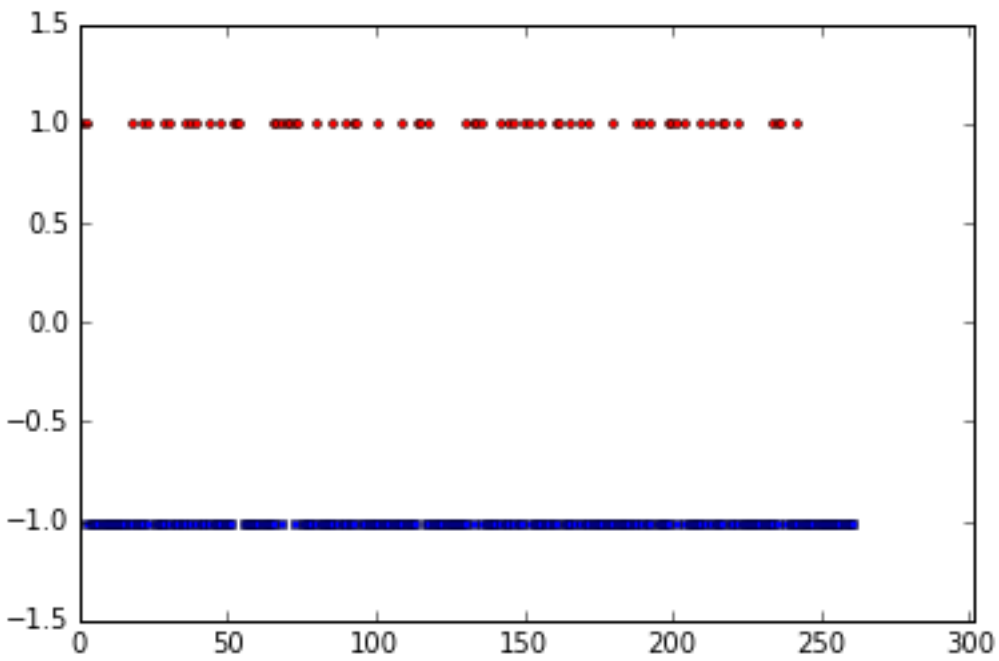
```
#  How Well Does the Linear Model Do?
p_linear = B_linear.dot(weights_linear)
p_linear[p_linear >  0] =  1
p_linear[p_linear <= 0] = -1
fp_linear = len(np.where(p_linear > v)[0])
fn_linear = len(np.where(p_linear < v)[0])
```

In [10]:

```
patientid = np.arange(1,len(p_linear)+1)
Blist = np.where(p_linear > 0)
Mlist = np.where(p_linear < 0)
plt.plot(patientid[Blist], p_linear[Blist], 'ro', ms=3, label='Benign')
plt.plot(patientid[Mlist], p_linear[Mlist], 'bs', ms=3, label='Malignant')
plt.axis([0,301,-1.5,1.5])
```

Out[10]:

```
[0, 301, -1.5, 1.5]
```

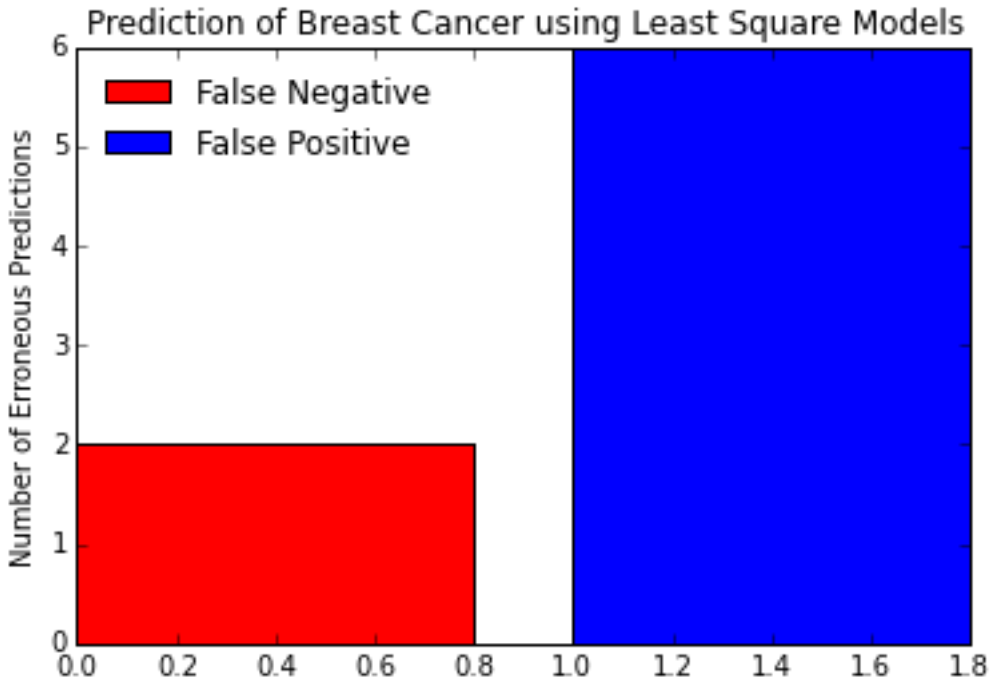


In [11]:

```
plt.bar(0,fn_linear, color='r', label='False Negative')
plt.bar(1,fp_linear, color='b', label='False Positive')
plt.ylabel('Number of Erroneous Predictions')
plt.title('Prediction of Breast Cancer using Least Square Models')
plt.legend(frameon=False, loc='upper left')
```

Out[11]:

```
<matplotlib.legend.Legend at 0x104320eb8>
```



In [27]:

```
numlinear = len(subset_labels)
numquad = len(subset_labels)
numcross = numlinear * (numlinear - 1) / 2
numcol = int(numlinear + numquad + numcross)
print(numcol)
A_quad = np.zeros((A_linear.shape[0], numcol))
B_quad = np.zeros((B_linear.shape[0], numcol))

# Linears
for i in range(numlinear):
    A_quad[:,i] = tumor_data[:, labelsid[subset_labels[i]]]
    B_quad[:,i] = validate_data[:, labelsid[subset_labels[i]]]

# Quadratics
for i in range(numquad):
    A_quad[:,i+numlinear] = tumor_data[:, labelsid[subset_labels[i]]]**2
    B_quad[:,i+numlinear] = validate_data[:, labelsid[subset_labels[i]]]**2

# Cross Terms
k = 0
for i, j in itertools.combinations([0,1,2,3], 2):
    A_quad[:,k+numlinear+numquad] =\
        tumor_data[:, labelsid[subset_labels[i]]]*tumor_data[:, labelsid[subset_
labels[j]]]
    B_quad[:,k+numlinear+numquad] =\
        validate_data[:, labelsid[subset_labels[i]]]*validate_data[:, labelsid[s
ubset_labels[j]]]
    k+=1
```

```
14
```

In [28]:

```
#  Solve the Quadratic Model
Q, R = la.qr(A_quad, "complete")
weights_quad = la.solve(R[:14], Q.T.dot(b)[:14])
```

In [29]:

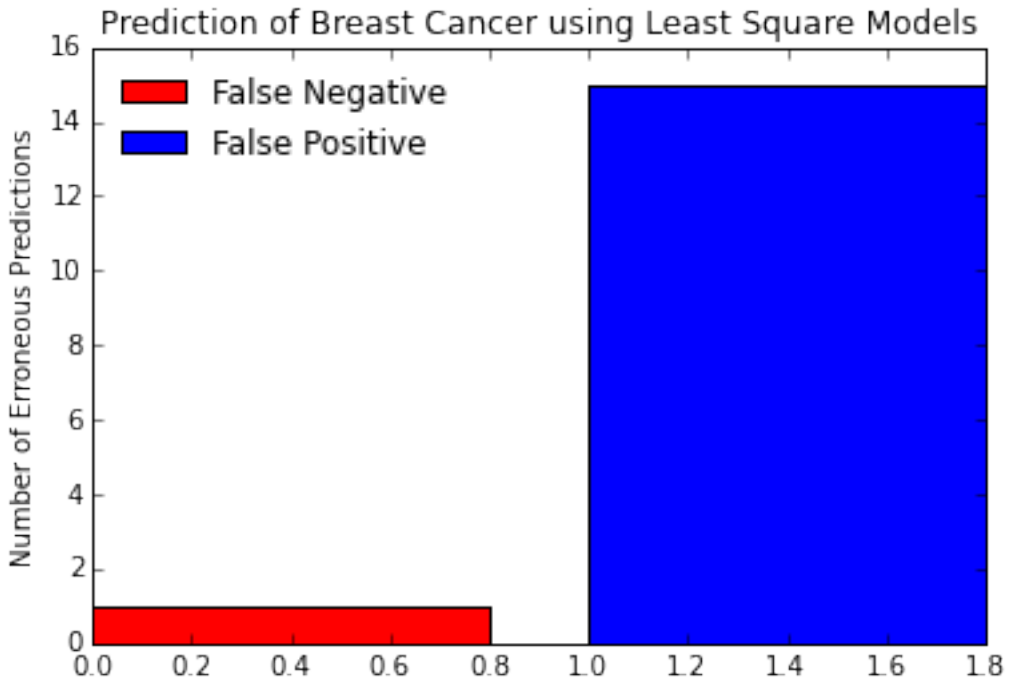
```
#  How Well Does the Quadratic Model Do?
p_quad = B_quad.dot(weights_quad)
p_quad[p_quad >  0] =  1
p_quad[p_quad <= 0] = -1
fp_quad = len(np.where(p_quad > v)[0])
fn_quad = len(np.where(p_quad < v)[0])
```

In [30]:

```
plt.bar(0,fn_quad, color='r', label='False Negative')
plt.bar(1,fp_quad, color='b', label='False Positive')
plt.ylabel('Number of Erroneous Predictions')
plt.title('Prediction of Breast Cancer using Least Square Models')
plt.legend(frameon=False, loc='upper left')
```

Out[30]:

```
<matplotlib.legend.Legend at 0x104319c88>
```



In [ ]: