In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from matplotlib import style

%matplotlib inline
```

In [2]:
```
df_ibm = pd.read_csv('IBM Attrition Data.csv')
```

In [3]:
```
df_ibm.head()
```

Out[3]:

	Age	Attrition	Department	DistanceFromHome	Education	EducationField	Environment
0	41	Yes	Sales	1	2	Life Sciences	
1	49	No	Research & Development	8	1	Life Sciences	
2	37	Yes	Research & Development	2	2	Other	
3	33	No	Research & Development	3	4	Life Sciences	
4	27	No	Research & Development	2	1	Medical	

In [4]:
```
df_ibm.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 13 columns):
Age                        1470 non-null int64
Attrition                  1470 non-null object
Department                 1470 non-null object
DistanceFromHome           1470 non-null int64
Education                  1470 non-null int64
EducationField             1470 non-null object
EnvironmentSatisfaction    1470 non-null int64
JobSatisfaction            1470 non-null int64
MaritalStatus              1470 non-null object
MonthlyIncome              1470 non-null int64
NumCompaniesWorked         1470 non-null int64
WorkLifeBalance            1470 non-null int64
YearsAtCompany             1470 non-null int64
dtypes: int64(9), object(4)
memory usage: 149.4+ KB
```

In [5]:
```
df_ibm.isna().sum()
```

Out[5]:
```
Age                         0
Attrition                   0
Department                  0
DistanceFromHome            0
Education                   0
EducationField              0
EnvironmentSatisfaction     0
JobSatisfaction             0
MaritalStatus               0
MonthlyIncome               0
NumCompaniesWorked          0
WorkLifeBalance             0
YearsAtCompany              0
dtype: int64
```

In [6]:
```
df_ibm['Department'].unique()
```

Out[6]:
```
array(['Sales', 'Research & Development', 'Human Resources'], dtyp
e=object)
```

In [7]:
```
df_ibm['EducationField'].unique()
```

Out[7]:
```
array(['Life Sciences', 'Other', 'Medical', 'Marketing',
       'Technical Degree', 'Human Resources'], dtype=object)
```

In [8]:
```
df_ibm['MaritalStatus'].unique()
```

Out[8]:
```
array(['Single', 'Married', 'Divorced'], dtype=object)
```

In [9]:
```
df_ibm['Department']= df_ibm['Department'].map({'Sales':1,'Research
& Development':2,'Human Resources':3})
```

In [10]:
```
df_ibm
```
Out[10]:

	Age	Attrition	Department	DistanceFromHome	Education	EducationField	Environm
0	41	Yes	1	1	2	Life Sciences	
1	49	No	2	8	1	Life Sciences	
2	37	Yes	2	2	2	Other	
3	33	No	2	3	4	Life Sciences	
4	27	No	2	2	1	Medical	
...	...	...	...	...	...	...	
1465	36	No	2	23	2	Medical	
1466	39	No	2	6	1	Medical	
1467	27	No	2	4	3	Life Sciences	
1468	49	No	1	2	3	Medical	
1469	34	No	2	8	3	Medical	

1470 rows × 13 columns

In [11]:
```
df_ibm['EducationField']= df_ibm['EducationField'].replace(['Life S
ciences', 'Other', 'Medical', 'Marketing',
       'Technical Degree', 'Human Resources'], [1,2,3,4,5,6])
```
In [12]:
```
df_ibm.head()
```
Out[12]:

	Age	Attrition	Department	DistanceFromHome	Education	EducationField	Environment
0	41	Yes	1	1	2	1	
1	49	No	2	8	1	1	
2	37	Yes	2	2	2	2	
3	33	No	2	3	4	1	
4	27	No	2	2	1	3	

In [13]:
```
marital_status = lambda x : 1 if (x=='Married') else (2 if x == 'Si
ngle' else 3)
```
In [14]:
```
df_ibm['MaritalStatus']= df_ibm['MaritalStatus'].map(marital_status
)
```

In [15]:
```
df_ibm.head()
```

Out[15]:

	Age	Attrition	Department	DistanceFromHome	Education	EducationField	Environment
0	41	Yes	1	1	2	1	
1	49	No	2	8	1	1	
2	37	Yes	2	2	2	2	
3	33	No	2	3	4	1	
4	27	No	2	2	1	3	

In [16]:
```
attrition = lambda x: 1 if(x=='Yes') else 0
```

In [17]:
```
df_ibm['Attrition']=df_ibm['Attrition'].map(attrition)
```

In [18]:
```
df_ibm.head()
```

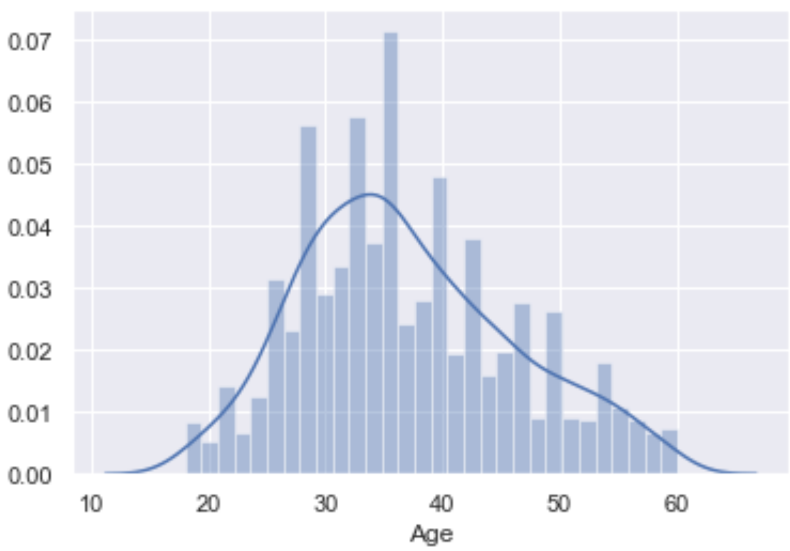
Out[18]:

	Age	Attrition	Department	DistanceFromHome	Education	EducationField	Environment
0	41	1	1	1	2	1	
1	49	0	2	8	1	1	
2	37	1	2	2	2	2	
3	33	0	2	3	4	1	
4	27	0	2	2	1	3	

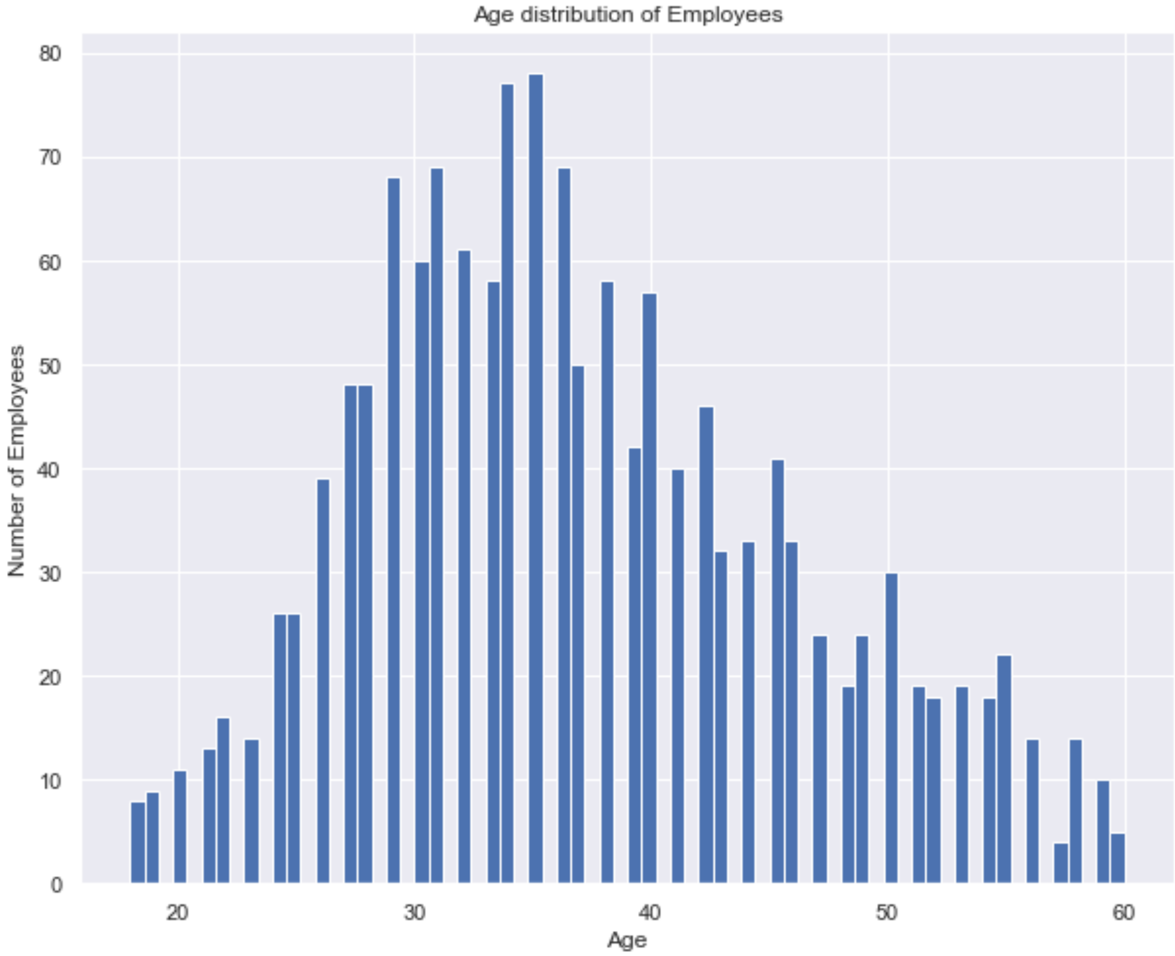
Age distribution of employees in IBM

In [19]:
```
sns.set(color_codes=True)
sns.distplot(df_ibm['Age'], bins=30)
```

Out[19]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x10fc82850>
```

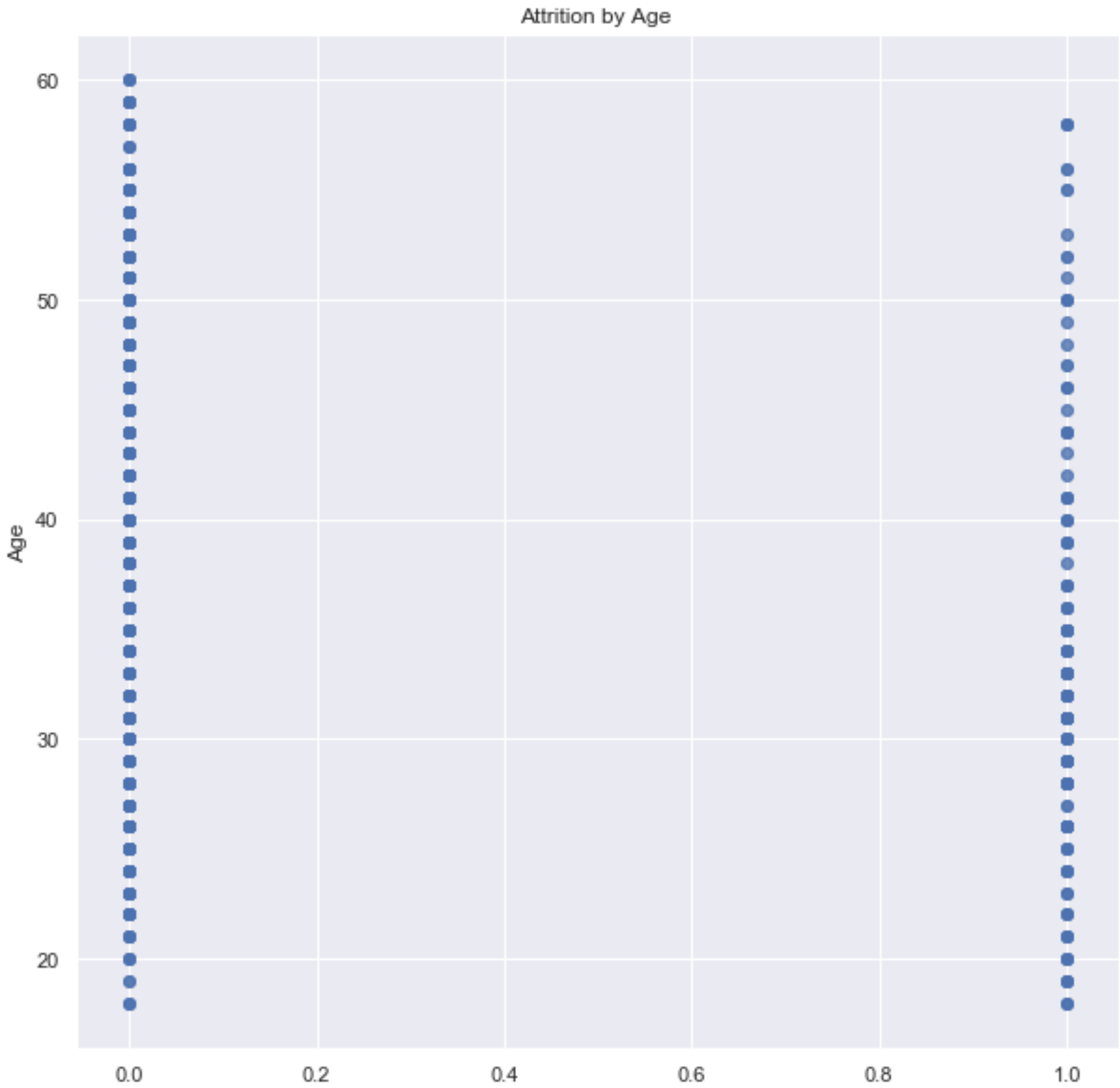


In [21]:
```
# histogram for age
plt.figure(figsize=(10,8))
df_ibm['Age'].hist(bins=70)
plt.title("Age distribution of Employees")
plt.xlabel("Age")
plt.ylabel("Number of Employees")
plt.show()
```



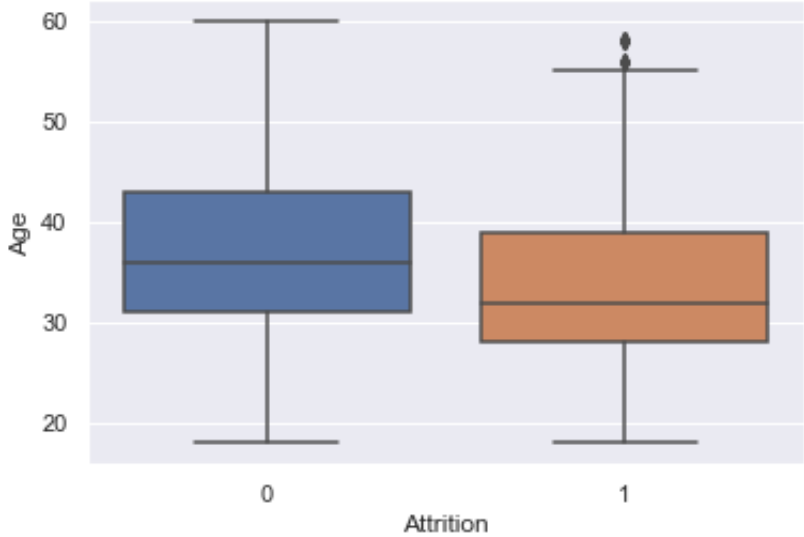
Explore attrition by age

In [23]:
```
plt.figure(figsize=(10,10))
plt.scatter(df_ibm.Attrition,df_ibm.Age, alpha=.55)
plt.title("Attrition by Age ")
plt.ylabel("Age")
plt.grid(b=True, which='major',axis='y')
plt.show()
```



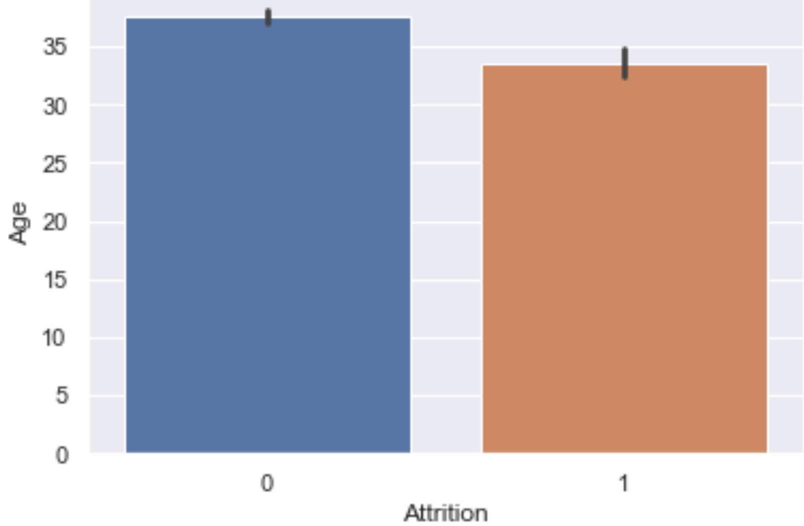
In [24]:
```
sns.boxplot(df_ibm['Attrition'], df_ibm['Age'])
```

Out[24]: <matplotlib.axes._subplots.AxesSubplot at 0x1a24c4eb50>



In [26]:
```
sns.barplot(df_ibm['Attrition'], df_ibm['Age'])
```

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x1a24a1ced0>

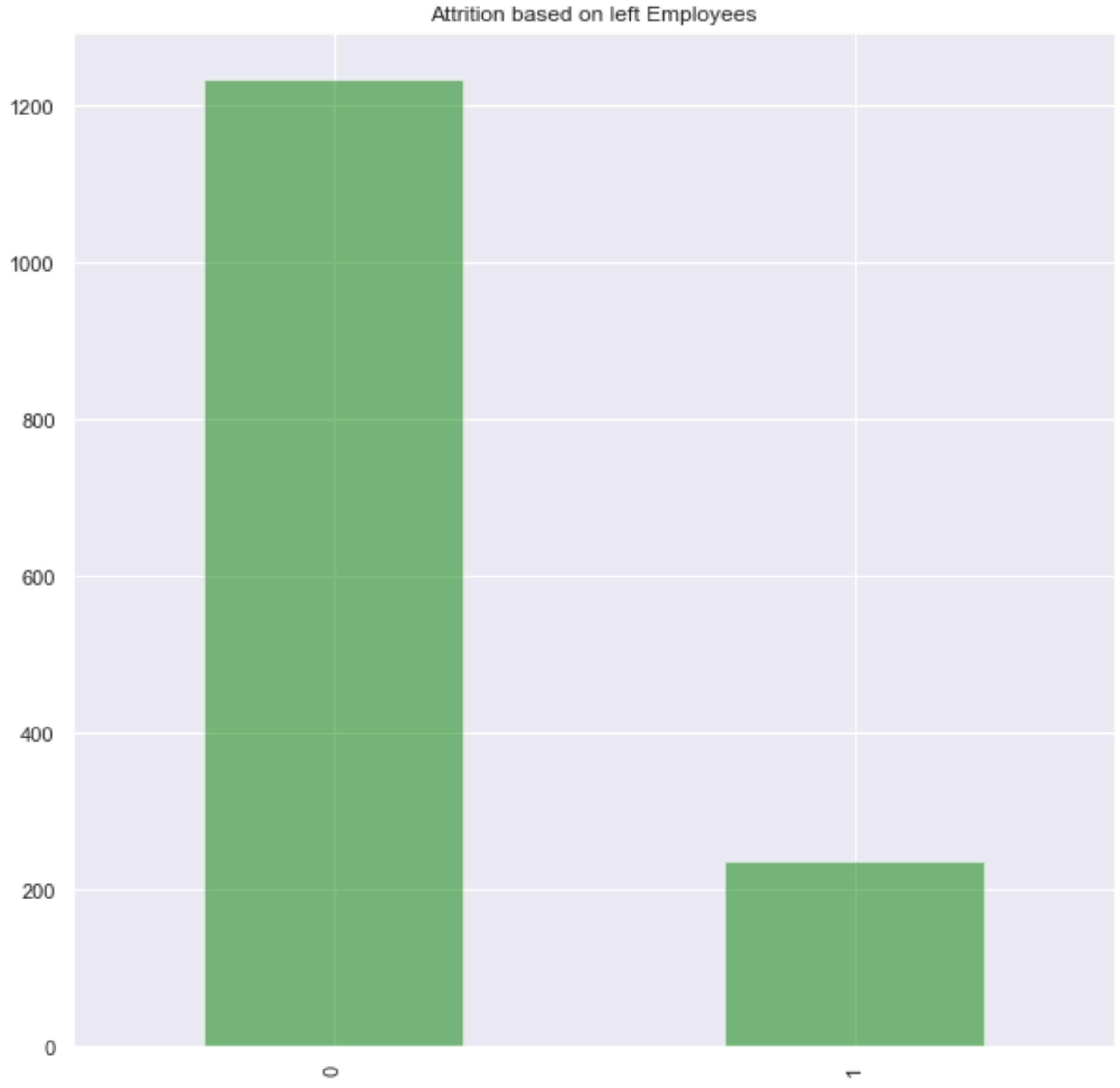


Explore data for Left employees

In [30]:
```
df_ibm.Attrition.value_counts()
```

Out[30]:
```
0    1233
1     237
Name: Attrition, dtype: int64
```

In [29]:
```
plt.figure(figsize=(10,10))
df_ibm.Attrition.value_counts().plot(kind='bar',color='green',alpha
=0.5)
plt.title('Attrition based on left Employees')
plt.show()
```



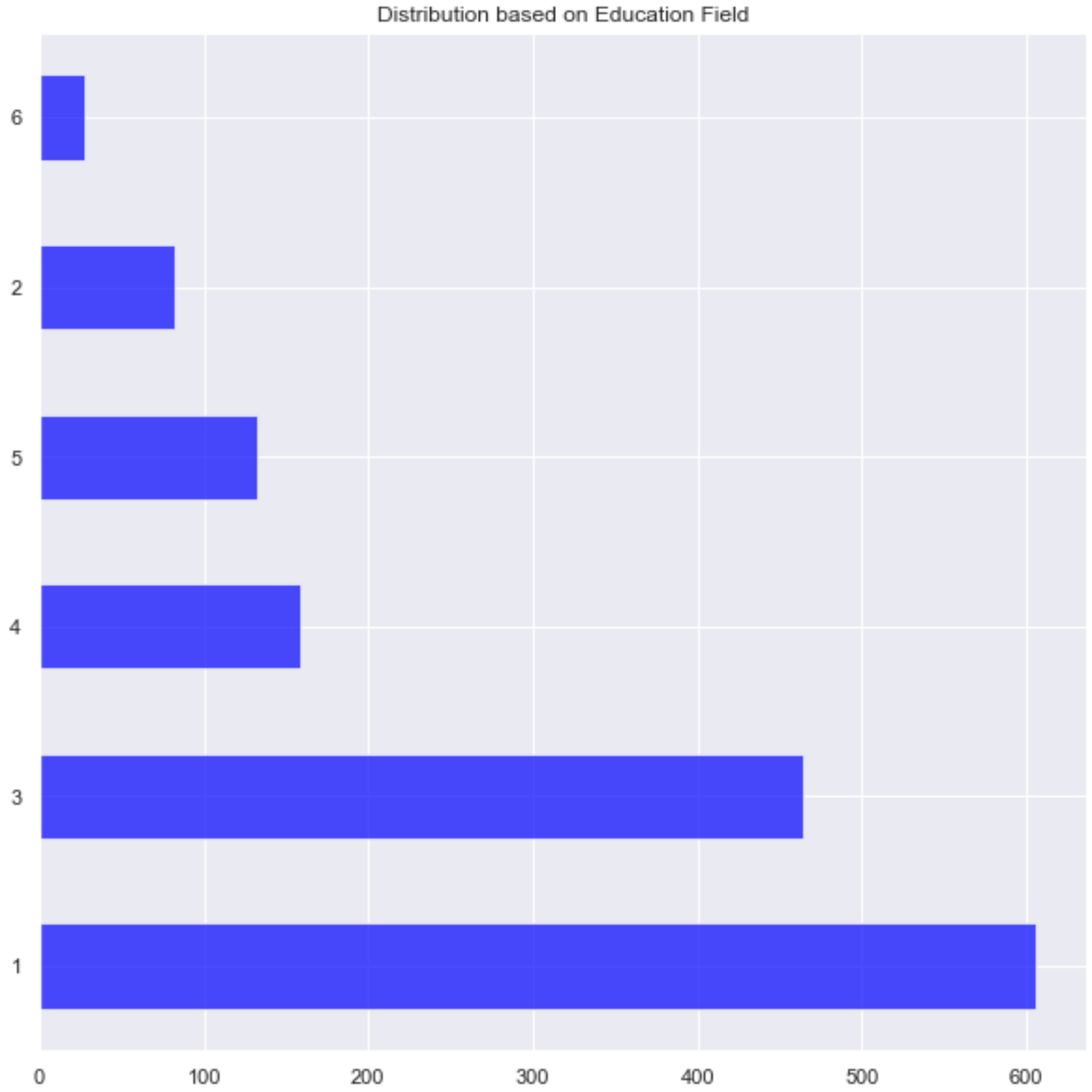
Find out the distribution of employees by the education field

In [31]:
```
df_ibm.EducationField.value_counts()
```

Out[31]:
```
1    606
3    464
4    159
5    132
2     82
6     27
Name: EducationField, dtype: int64
```

In [32]:

```
plt.figure(figsize=(10,10))
df_ibm.EducationField.value_counts().plot(kind='barh',color='blue',
alpha=0.7)
plt.title('Distribution based on Education Field')
plt.show()
```



Give a bar chart for the number of married and unmarried employees

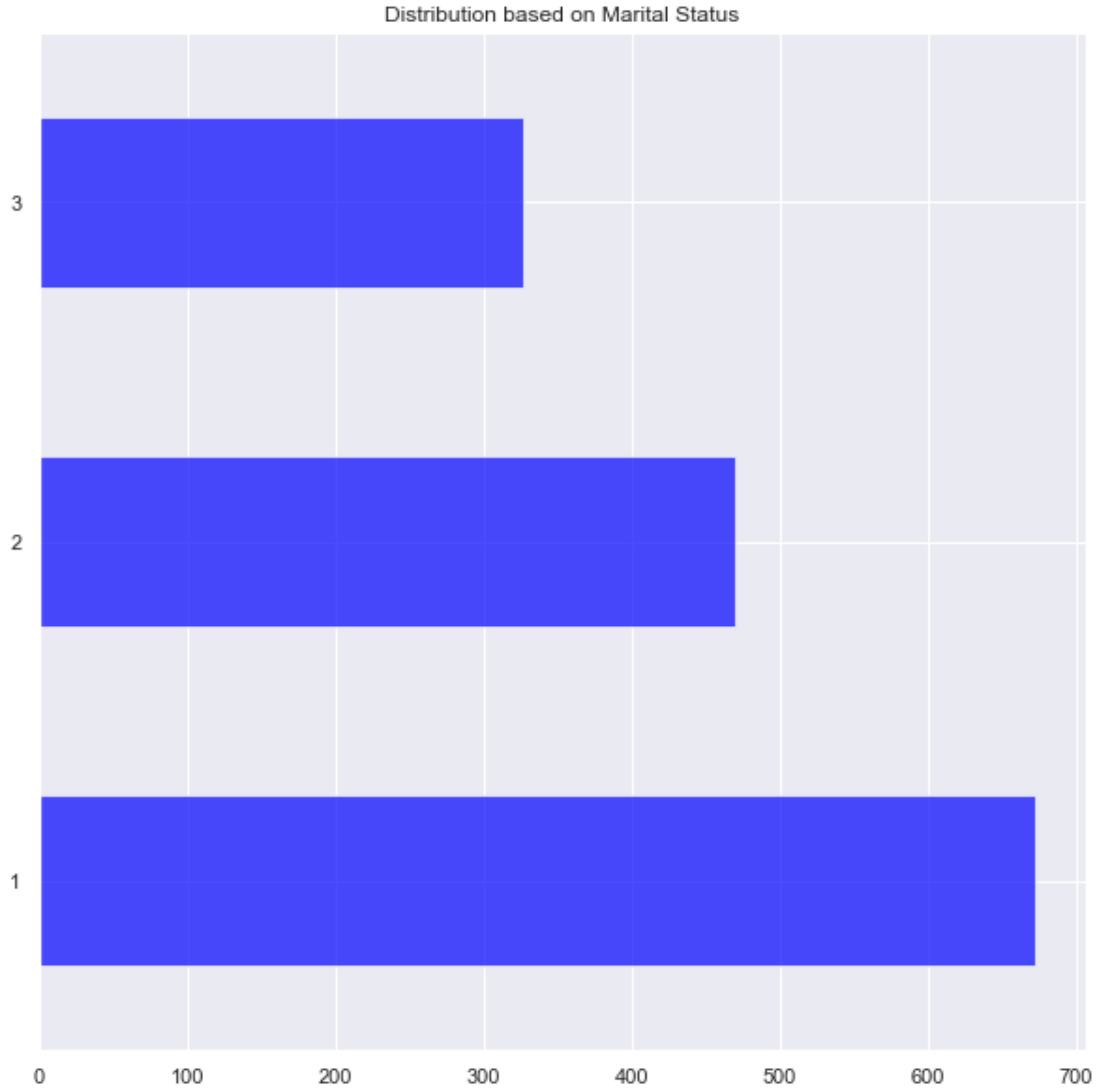
In [33]:

```
df_ibm.MaritalStatus.value_counts()
```

Out[33]:

```
1    673
2    470
3    327
Name: MaritalStatus, dtype: int64
```

In [34]:
```
plt.figure(figsize=(10,10))
df_ibm.MaritalStatus.value_counts().plot(kind='barh',color='blue',a
lpha=0.7)
plt.title('Distribution based on Marital Status')
plt.show()
```



Build up a logistic regression model to predict which employees are likely to attrite.

In [36]:
```
df_ibm.describe()
```

Out[36]:

	Age	Attrition	Department	DistanceFromHome	Education	EducationI
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	1470.00
mean	36.923810	0.161224	1.739456	9.192517	2.912925	2.46
std	9.135373	0.367863	0.527792	8.106864	1.024165	1.43
min	18.000000	0.000000	1.000000	1.000000	1.000000	1.00
25%	30.000000	0.000000	1.000000	2.000000	2.000000	1.00
50%	36.000000	0.000000	2.000000	7.000000	3.000000	3.00
75%	43.000000	0.000000	2.000000	14.000000	4.000000	3.00
max	60.000000	1.000000	3.000000	29.000000	5.000000	6.00

In [37]:
```
#Building the model
X = df_ibm.drop('Attrition',axis=1)
Y = df_ibm['Attrition']
```

In [38]:
```
X.head()
```

Out[38]:

	Age	Department	DistanceFromHome	Education	EducationField	EnvironmentSatisfactic
0	41	1	1	2	1	
1	49	2	8	1	1	
2	37	2	2	2	2	
3	33	2	3	4	1	
4	27	2	2	1	3	

In [39]:
```
Y.head()
```

Out[39]:
```
0    1
1    0
2    1
3    0
4    0
Name: Attrition, dtype: int64
```

In [40]:
```
X.dtypes
```

Out[40]:
```
Age                        int64
Department                 int64
DistanceFromHome           int64
Education                  int64
EducationField             int64
EnvironmentSatisfaction    int64
JobSatisfaction            int64
MaritalStatus              int64
MonthlyIncome              int64
NumCompaniesWorked         int64
WorkLifeBalance            int64
YearsAtCompany             int64
dtype: object
```

In [41]:
```
Y.dtypes
```

Out[41]:
```
dtype('int64')
```

In [42]:
```
from sklearn.linear_model import LogisticRegression

model = LogisticRegression()
model = model.fit(X, Y)

# check the accuracy on the training set
model.score(X, Y)
```

```
/Users/rgm/Python/anaconda3/lib/python3.7/site-packages/sklearn/li
near_model/logistic.py:432: FutureWarning: Default solver will be
changed to 'lbfgs' in 0.22. Specify a solver to silence this warni
ng.
  FutureWarning)
```

Out[42]:
```
0.8414965986394558
```

In [43]:
```
Y.mean()
```

Out[43]:
```
0.16122448979591836
```

In [46]:
```python
from sklearn.model_selection import train_test_split

X_train,X_test,y_train,y_test=train_test_split(X,Y, test_size=0.3,
random_state=0)
model_log=LogisticRegression()
model_log.fit(X_train, y_train)
```

```
/Users/rgm/Python/anaconda3/lib/python3.7/site-packages/sklearn/li
near_model/logistic.py:432: FutureWarning: Default solver will be
changed to 'lbfgs' in 0.22. Specify a solver to silence this warni
ng.
  FutureWarning)
```

Out[46]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_inter
cept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=10
0,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, v
erbose=0,
                   warm_start=False)
```

In [47]:
```python
predicted= model_log.predict(X_test)
print (predicted)
```

```
[0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 1
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0
 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0]
```

In [48]:
```python
probs = model_log.predict_proba(X_test)
print (probs)
```

```
[[0.88884691 0.11115309]
 [0.85804    0.14196   ]
 [0.80320187 0.19679813]
```

```
[0.8046951  0.1953049 ]
[0.84475259 0.15524741]
[0.79274143 0.20725857]
[0.73667156 0.26332844]
[0.74416788 0.25583212]
[0.97800501 0.02199499]
[0.82782481 0.17217519]
[0.97451731 0.02548269]
[0.76990317 0.23009683]
[0.93608341 0.06391659]
[0.72959139 0.27040861]
[0.83974088 0.16025912]
[0.89279352 0.10720648]
[0.93158795 0.06841205]
[0.91307024 0.08692976]
[0.85613082 0.14386918]
[0.71893295 0.28106705]
[0.78265545 0.21734455]
[0.95567659 0.04432341]
[0.89010586 0.10989414]
[0.93498953 0.06501047]
[0.56542427 0.43457573]
[0.82497249 0.17502751]
[0.84647082 0.15352918]
[0.95019302 0.04980698]
[0.72298864 0.27701136]
[0.88093912 0.11906088]
[0.90555625 0.09444375]
[0.8260889  0.1739111 ]
[0.84489875 0.15510125]
[0.89652765 0.10347235]
[0.94940822 0.05059178]
[0.92906341 0.07093659]
[0.92810718 0.07189282]
[0.89093261 0.10906739]
[0.94645849 0.05354151]
[0.85290808 0.14709192]
[0.92087043 0.07912957]
[0.86970393 0.13029607]
[0.94619965 0.05380035]
[0.93476726 0.06523274]
[0.89979262 0.10020738]
[0.8143875  0.1856125 ]
[0.62569622 0.37430378]
[0.93197186 0.06802814]
[0.61875823 0.38124177]
[0.7809789  0.2190211 ]
[0.96301726 0.03698274]
[0.65624639 0.34375361]
[0.8039433  0.1960567 ]
[0.62856369 0.37143631]
[0.83196197 0.16803803]
[0.80065294 0.19934706]
[0.86934961 0.13065039]
[0.83928024 0.16071976]
[0.84951632 0.15048368]
[0.63776278 0.36223722]
```

```
[0.94599714 0.05400286]
[0.79178107 0.20821893]
[0.9052715  0.0947285 ]
[0.86670738 0.13329262]
[0.7765121  0.2234879 ]
[0.91184807 0.08815193]
[0.85185175 0.14814825]
[0.70155429 0.29844571]
[0.85754986 0.14245014]
[0.68700121 0.31299879]
[0.86704388 0.13295612]
[0.74065761 0.25934239]
[0.82873071 0.17126929]
[0.87320314 0.12679686]
[0.72837714 0.27162286]
[0.81958538 0.18041462]
[0.87810998 0.12189002]
[0.9824641  0.0175359 ]
[0.83974448 0.16025552]
[0.8468493  0.1531507 ]
[0.9770752  0.0229248 ]
[0.83375543 0.16624457]
[0.75098222 0.24901778]
[0.91343287 0.08656713]
[0.9117861  0.0882139 ]
[0.78702835 0.21297165]
[0.96377662 0.03622338]
[0.90012817 0.09987183]
[0.88693649 0.11306351]
[0.8605043  0.1394957 ]
[0.55795601 0.44204399]
[0.72324042 0.27675958]
[0.64506118 0.35493882]
[0.7883582  0.2116418 ]
[0.92711762 0.07288238]
[0.82529368 0.17470632]
[0.71688251 0.28311749]
[0.6510961  0.3489039 ]
[0.86559145 0.13440855]
[0.93962728 0.06037272]
[0.53730943 0.46269057]
[0.64310092 0.35689908]
[0.97444122 0.02555878]
[0.92122016 0.07877984]
[0.82170785 0.17829215]
[0.88386995 0.11613005]
[0.92872849 0.07127151]
[0.83570464 0.16429536]
[0.76943307 0.23056693]
[0.92937834 0.07062166]
[0.77786302 0.22213698]
[0.9845391  0.0154609 ]
[0.95169648 0.04830352]
[0.83117478 0.16882522]
[0.79386838 0.20613162]
[0.9235745  0.0764255 ]
[0.95864137 0.04135863]
```

```
[0.96799739 0.03200261]
[0.97134943 0.02865057]
[0.93121017 0.06878983]
[0.73934598 0.26065402]
[0.9022203  0.0977797 ]
[0.91284585 0.08715415]
[0.72527433 0.27472567]
[0.95653051 0.04346949]
[0.95369388 0.04630612]
[0.97816952 0.02183048]
[0.85567342 0.14432658]
[0.87556815 0.12443185]
[0.92200282 0.07799718]
[0.96122269 0.03877731]
[0.80809692 0.19190308]
[0.84968385 0.15031615]
[0.70508108 0.29491892]
[0.95869189 0.04130811]
[0.84898479 0.15101521]
[0.81196127 0.18803873]
[0.88087132 0.11912868]
[0.73472185 0.26527815]
[0.82172673 0.17827327]
[0.93903385 0.06096615]
[0.90694855 0.09305145]
[0.90975775 0.09024225]
[0.95952302 0.04047698]
[0.80649109 0.19350891]
[0.87412787 0.12587213]
[0.97871833 0.02128167]
[0.98429175 0.01570825]
[0.97713555 0.02286445]
[0.81622286 0.18377714]
[0.88848774 0.11151226]
[0.86081068 0.13918932]
[0.78934733 0.21065267]
[0.96157238 0.03842762]
[0.94216406 0.05783594]
[0.95378841 0.04621159]
[0.98107117 0.01892883]
[0.71439781 0.28560219]
[0.76117614 0.23882386]
[0.92729048 0.07270952]
[0.96304425 0.03695575]
[0.94948781 0.05051219]
[0.59577147 0.40422853]
[0.67537202 0.32462798]
[0.95731056 0.04268944]
[0.89676311 0.10323689]
[0.90245983 0.09754017]
[0.84976919 0.15023081]
[0.84666801 0.15333199]
[0.98842456 0.01157544]
[0.90506776 0.09493224]
[0.92320003 0.07679997]
[0.87435025 0.12564975]
[0.67115185 0.32884815]
```

```
[0.96813074 0.03186926]
[0.88421981 0.11578019]
[0.85538942 0.14461058]
[0.96964451 0.03035549]
[0.67420056 0.32579944]
[0.95770756 0.04229244]
[0.80648546 0.19351454]
[0.87445855 0.12554145]
[0.97480231 0.02519769]
[0.75757208 0.24242792]
[0.89152163 0.10847837]
[0.77628368 0.22371632]
[0.57469817 0.42530183]
[0.72271399 0.27728601]
[0.8451573  0.1548427 ]
[0.63953676 0.36046324]
[0.63119415 0.36880585]
[0.76114812 0.23885188]
[0.8825116  0.1174884 ]
[0.88642068 0.11357932]
[0.8904538  0.1095462 ]
[0.92603009 0.07396991]
[0.93675366 0.06324634]
[0.86774277 0.13225723]
[0.88320408 0.11679592]
[0.8784075  0.1215925 ]
[0.81782527 0.18217473]
[0.81594257 0.18405743]
[0.7352273  0.2647727 ]
[0.86915473 0.13084527]
[0.66244273 0.33755727]
[0.84843142 0.15156858]
[0.86193743 0.13806257]
[0.97684776 0.02315224]
[0.75977291 0.24022709]
[0.95572509 0.04427491]
[0.73878786 0.26121214]
[0.78109595 0.21890405]
[0.88040917 0.11959083]
[0.66184134 0.33815866]
[0.88575118 0.11424882]
[0.90478959 0.09521041]
[0.94667552 0.05332448]
[0.86257301 0.13742699]
[0.97063701 0.02936299]
[0.72472045 0.27527955]
[0.58296785 0.41703215]
[0.75481261 0.24518739]
[0.70233556 0.29766444]
[0.98574274 0.01425726]
[0.85431658 0.14568342]
[0.87772128 0.12227872]
[0.87090068 0.12909932]
[0.81382269 0.18617731]
[0.65140702 0.34859298]
[0.89672611 0.10327389]
[0.98786458 0.01213542]
```

```
[0.91993892 0.08006108]
[0.91008542 0.08991458]
[0.92764484 0.07235516]
[0.79946224 0.20053776]
[0.90427718 0.09572282]
[0.56323943 0.43676057]
[0.8351812  0.1648188 ]
[0.78708323 0.21291677]
[0.96702566 0.03297434]
[0.84236157 0.15763843]
[0.8095116  0.1904884 ]
[0.98478562 0.01521438]
[0.88138935 0.11861065]
[0.80580635 0.19419365]
[0.93321366 0.06678634]
[0.88982218 0.11017782]
[0.83575013 0.16424987]
[0.86348141 0.13651859]
[0.8954566  0.1045434 ]
[0.78446709 0.21553291]
[0.88700037 0.11299963]
[0.41371642 0.58628358]
[0.94223046 0.05776954]
[0.8883312  0.1116688 ]
[0.74514254 0.25485746]
[0.99096955 0.00903045]
[0.75379666 0.24620334]
[0.44685442 0.55314558]
[0.61090979 0.38909021]
[0.76740824 0.23259176]
[0.89073    0.10927   ]
[0.9077099  0.0922901 ]
[0.85957293 0.14042707]
[0.71478893 0.28521107]
[0.8653085  0.1346915 ]
[0.91800789 0.08199211]
[0.82776731 0.17223269]
[0.53431746 0.46568254]
[0.88790465 0.11209535]
[0.81238102 0.18761898]
[0.90444532 0.09555468]
[0.75740588 0.24259412]
[0.83500167 0.16499833]
[0.83281347 0.16718653]
[0.92733845 0.07266155]
[0.8440804  0.1559196 ]
[0.93166238 0.06833762]
[0.72923662 0.27076338]
[0.89873905 0.10126095]
[0.69275209 0.30724791]
[0.92581078 0.07418922]
[0.68910676 0.31089324]
[0.91195444 0.08804556]
[0.99320944 0.00679056]
[0.93196207 0.06803793]
[0.87964212 0.12035788]
[0.85779361 0.14220639]
```

```
[0.87460611 0.12539389]
[0.90526495 0.09473505]
[0.92945995 0.07054005]
[0.84674073 0.15325927]
[0.89522273 0.10477727]
[0.88869282 0.11130718]
[0.97727217 0.02272783]
[0.74738473 0.25261527]
[0.76942751 0.23057249]
[0.85045098 0.14954902]
[0.78869611 0.21130389]
[0.98732489 0.01267511]
[0.83354769 0.16645231]
[0.81802171 0.18197829]
[0.83913449 0.16086551]
[0.93130917 0.06869083]
[0.88316961 0.11683039]
[0.9439371  0.0560629 ]
[0.82797644 0.17202356]
[0.46342677 0.53657323]
[0.95824465 0.04175535]
[0.8543708  0.1456292 ]
[0.87159837 0.12840163]
[0.88033211 0.11966789]
[0.88190062 0.11809938]
[0.82665446 0.17334554]
[0.68867442 0.31132558]
[0.69303889 0.30696111]
[0.83415732 0.16584268]
[0.87179515 0.12820485]
[0.53870727 0.46129273]
[0.91951477 0.08048523]
[0.71392513 0.28607487]
[0.95491331 0.04508669]
[0.82669324 0.17330676]
[0.72906226 0.27093774]
[0.90252324 0.09747676]
[0.95067716 0.04932284]
[0.72898008 0.27101992]
[0.91905959 0.08094041]
[0.99020649 0.00979351]
[0.88648824 0.11351176]
[0.89971016 0.10028984]
[0.85990883 0.14009117]
[0.948687   0.051313  ]
[0.86282297 0.13717703]
[0.82520288 0.17479712]
[0.89928162 0.10071838]
[0.93481632 0.06518368]
[0.95519135 0.04480865]
[0.842554   0.157446  ]
[0.95276664 0.04723336]
[0.77452803 0.22547197]
[0.83941066 0.16058934]
[0.82202544 0.17797456]
[0.69093192 0.30906808]
[0.91168817 0.08831183]
```

```
[0.85605744 0.14394256]
[0.79159386 0.20840614]
[0.97983941 0.02016059]
[0.63618316 0.36381684]
[0.76779169 0.23220831]
[0.82262142 0.17737858]
[0.66651297 0.33348703]
[0.85492068 0.14507932]
[0.85015012 0.14984988]
[0.8280353  0.1719647 ]
[0.87182884 0.12817116]
[0.83863239 0.16136761]
[0.86439346 0.13560654]
[0.95797295 0.04202705]
[0.91673023 0.08326977]
[0.98551088 0.01448912]
[0.92368175 0.07631825]
[0.92199007 0.07800993]
[0.89286724 0.10713276]
[0.77666263 0.22333737]
[0.90518468 0.09481532]
[0.97476387 0.02523613]
[0.70401669 0.29598331]
[0.95578101 0.04421899]
[0.86871537 0.13128463]
[0.94215167 0.05784833]
[0.81303043 0.18696957]
[0.91115625 0.08884375]
[0.93524969 0.06475031]
[0.76826904 0.23173096]
[0.93014251 0.06985749]
[0.94293966 0.05706034]
[0.93355314 0.06644686]
[0.81314012 0.18685988]
[0.94034721 0.05965279]
[0.89205822 0.10794178]
[0.88164701 0.11835299]
[0.92066581 0.07933419]
[0.91122317 0.08877683]
[0.98029575 0.01970425]
[0.45464657 0.54535343]
[0.94128237 0.05871763]
[0.87284043 0.12715957]
[0.8941476  0.1058524 ]
[0.78369141 0.21630859]
[0.88790125 0.11209875]
[0.95678756 0.04321244]
[0.75342415 0.24657585]
[0.88147514 0.11852486]
[0.50678614 0.49321386]
[0.85542594 0.14457406]
[0.87019527 0.12980473]
[0.65214575 0.34785425]
[0.93116328 0.06883672]
[0.94517672 0.05482328]
[0.64870696 0.35129304]
[0.76380794 0.23619206]
```

```
 [0.95891084 0.04108916]
 [0.73571524 0.26428476]
 [0.86980074 0.13019926]
 [0.9218691  0.0781309 ]
 [0.869738   0.130262  ]
 [0.79180457 0.20819543]
 [0.93590066 0.06409934]
 [0.93205733 0.06794267]
 [0.67648037 0.32351963]
 [0.97785398 0.02214602]
 [0.81155781 0.18844219]
 [0.82849712 0.17150288]
 [0.58729339 0.41270661]
 [0.77529154 0.22470846]
 [0.95058721 0.04941279]
 [0.91208513 0.08791487]
 [0.92867276 0.07132724]
 [0.91154312 0.08845688]
 [0.87203815 0.12796185]
 [0.6018556  0.3981444 ]
 [0.91364102 0.08635898]
 [0.98457011 0.01542989]
 [0.91681352 0.08318648]
 [0.84768668 0.15231332]
 [0.8084883  0.1915117 ]
 [0.83467082 0.16532918]
 [0.74726658 0.25273342]
 [0.9603029  0.0396971 ]
 [0.76330571 0.23669429]
 [0.68832933 0.31167067]
 [0.93164104 0.06835896]
 [0.6564933  0.3435067 ]
 [0.85577709 0.14422291]
 [0.93032546 0.06967454]
 [0.88656215 0.11343785]
 [0.90522754 0.09477246]
 [0.79435162 0.20564838]
 [0.95575092 0.04424908]
 [0.96089678 0.03910322]]
```

In [49]:

```
from sklearn import metrics

print (metrics.accuracy_score(y_test, predicted))
print (metrics.roc_auc_score(y_test, probs[:, 1]))
```

```
0.8458049886621315
0.6949942241047362
```

In [50]:
```
print (metrics.confusion_matrix(y_test, predicted))
print (metrics.classification_report(y_test, predicted))
```

```
[[370   1]
 [ 67   3]]
              precision    recall  f1-score   support

           0       0.85      1.00      0.92       371
           1       0.75      0.04      0.08        70

    accuracy                           0.85       441
   macro avg       0.80      0.52      0.50       441
weighted avg       0.83      0.85      0.78       441
```