KNN in Python

In [15]:
```
import pandas as pd
import numpy as np
import os
from sklearn.model_selection import RepeatedKFold, cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
```

In [6]:
```
os.chdir("C:\\Users\\Matt\\Documents\\Python_Projects")

baseball_train = pd.read_csv(r"baseball_train.csv",index_col=0,
 dtype={'Opp': 'category', 'Result': 'category',
'Name': 'category'}, header=0)
baseball_test = pd.read_csv(r"baseball_test.csv",index_col=0,
 dtype={'Opp': 'category', 'Result': 'category', 'Name': 'category'}, header=0
)
```

In [7]:
```
X = baseball_train.iloc[:,:-1]
X = X[['H','R','ERA','BB','SO','GB','FB','LD','PO','PU','Unk','SB','IBB']]
y = baseball_train.iloc[:,-1]
```

In [8]:
```
rkf = RepeatedKFold(n_splits=5, n_repeats=10, random_state=21191)
```

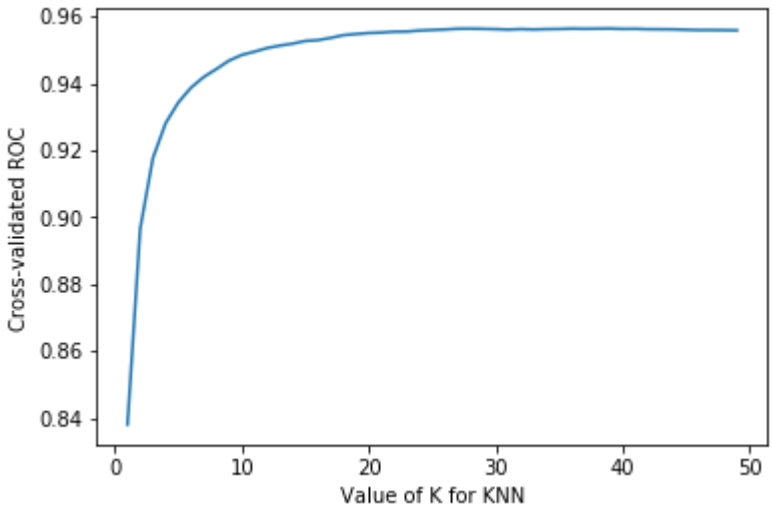
In [13]:
```
# range of k we want to try
k_range = range(1, 50)
# empty list to store scores
k_scores = []

# 1. we will loop through reasonable values of k
for k in k_range:
    # 2. run KNeighborsClassifier with k neighbours
    knn = KNeighborsClassifier(n_neighbors=k)
    # 3. obtain cross_val_score for KNeighborsClassifier with k neighbours
    cv_results = cross_val_score(knn,
                                 X,
                                 y,
                                 cv=rkf,
                                 scoring="roc_auc")
    # 4. append mean of scores for k neighbors to k_scores list
    k_scores.append(cv_results.mean())
```

In [11]:
```
# plot how accuracy changes as we vary k
import matplotlib.pyplot as plt
%matplotlib inline

# plot the value of K for KNN (x-axis) versus the cross-validated accuracy (y-axis)
# plt.plot(x_axis, y_axis)
plt.plot(k_range, k_scores)
plt.xlabel('Value of K for KNN')
plt.ylabel('Cross-validated ROC')
```

Out[11]:
```
Text(0, 0.5, 'Cross-validated ROC')
```



In [17]:
```
Xnew = baseball_test.iloc[:,:-1]
Xnew = Xnew[['H','R','ERA','BB','SO','GB','FB','LD','PO','PU','Unk','SB','IBB'
]]
yTrue = baseball_test.iloc[:,-1]

# check classification accuracy of KNN with K=7
knn = KNeighborsClassifier(n_neighbors=7)
knn.fit(X, y)
ynew = knn.predict(Xnew)
baseball = {'predicted': ynew, 'truth': yTrue}
print(pd.DataFrame(data=baseball))
metrics.accuracy_score(yTrue, ynew)
```

```
     predicted  truth
788      Nolan  Nolan
1463     Tommy  Tommy
1272     Nolan  Tommy
639      Nolan  Nolan
41       Tommy  Nolan
391      Nolan  Nolan
779      Nolan  Nolan
1457     Tommy  Tommy
496      Nolan  Nolan
678      Nolan  Nolan
358      Nolan  Nolan
67       Tommy  Nolan
1185     Tommy  Tommy
1096     Nolan  Tommy
946      Nolan  Tommy
911      Nolan  Tommy
1542     Tommy  Tommy
324      Nolan  Nolan
955      Tommy  Tommy
206      Nolan  Nolan
```

Out[17]: 0.7