20122675 박동현 화합물빅데이터 과제

목표: 원자 수와 본드 수의 correlataion analysis

```
f=open('/share/class/pubchem.sdf')
fo = open('pubchem_my.sdf', 'w')
def atom_and_bond_counter(sdf):
    atomid=sdf.split('\n')[0]
    atomline=sdf.split('\n')[3]
    atom=atomline.split()[0]
    bond=atomline.split()[1]
    return int(atom), int(bond)

cnt = 0
sdf = ''
atom_bond = []
for line in f:
    sdf=sdf+line
    if '$$$$' in line:
        cnt = cnt + 1
        sd = sdf.split('M  END')[0]
        sd = sd + 'M  END\n' + '$$$$\n'
        fo.write(sd)
        atom_bond.append([atom_and_bond_counter(sdf)])
        sdf=''

print atom_bond
```

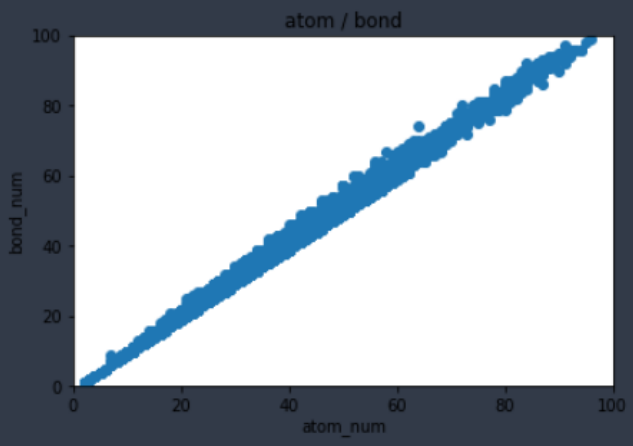
Atom개수와 bond수를 list로 뽑는다.

```
In [28]:
1  atom_bond = [[(31, 30)], [(32, 31)], [(19, 19)], [(14, 13)], [(1
2

In [34]:
1  atom_bond_pre = [x for x in atom_bond if not x[0][0] > 80000]
2  atom = [x[0][0] for x in atom_bond_pre]
3  bond = [x[0][1] for x in atom_bond_pre]

In [35]:
1  print(len(atom_bond))
2  print(len(atom_bond_pre))

18149
18136
```

주피터 환경에서 atom_bond를 가져와 atom_num이 80000이 넘어가는 원자들을 전처리 해준 후

In [42]:

```
from matplotlib import pyplot as plt
%matplotlib inline

plt.scatter(atom, bond)
plt.title("atom / bond")
plt.xlabel("atom_num")
plt.ylabel("bond_num")
plt.axis([0, 100, 0, 100])
plt.show()
```



Matplot을 통해 시각화 해준다. 그래프가 1자로 잘 나왔다.

In [56]:

```
from scipy import stats
import numpy as np

#x = np.random.random(10)
#y = np.random.random(10)
slope, intercept, r_value, p_value, std_err = stats.linregress(atom,bond)
```

In [57]:

```
print(slope)
print(intercept)
print(r_value)
print(std_err)
```

```
1.0579378298457003
-1.1857163670648454
0.9975451057039016
0.0005514999721637757
```

In [58]:

```
print("r-squared:", r_value**2)
```

```
r-squared: 0.9950962379138083
```

r-squared값이 거의 1에 가깝게 나왔다. 이를 통해 본드와 원자의 상관관계는 높다고 볼 수 있다.