In [71]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import warnings
import xgboost as xgb
warnings.filterwarnings('ignore')

from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.model_selection import GridSearchCV
```

In [72]:

```
ccdict = {}
letter = "this is the mid term report"

for c in letter:
    if not c in ccdict:
        ccdict[c] = 1
    else:
        ccdict[c] += 1
```

In [73]:

```
print(ccdict)
```

```
{'t': 4, 'h': 2, 'i': 3, 's': 2, ' ': 5, 'e': 3, 'm': 2, 'd': 1, 'r': 3, 'p': 1, 'o': 1}
```

In [74]:

```
def maxchar(letter):
    ccdict ={}
    for c in letter:
        if not c in ccdict:
            ccdict[c] = 1
        else:
            ccdict[c] += 1
    return(max(ccdict.items()))
```

In [75]:

```
maxchar("this is the mid term report")
```

Out[75]:

```
('t', 4)
```

In [80]:

```
df = pd.read_csv("tokyo_monthly_average_max_1950_2017.csv",header=None)
```

In [105]:

```
tempvec = np.array(df[1])
```

In [110]:

```
temp_mat = tempvec.reshape(68,12)
```

In [120]:

```
temp_df = pd.DataFrame(temp_mat)
```

In [121]:

```
temp_df.head()
```

Out[121]:

| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 5.0 | 4.7 | 7.7 | 13.9 | 19.2 | 21.8 | 26.5 | 26.2 | 23.8 | 15.8 | 11.1 | 5.4 |
| **1** | 3.3 | 4.5 | 8.8 | 13.3 | 18.0 | 21.2 | 24.3 | 26.7 | 20.7 | 17.3 | 11.4 | 7.3 |
| **2** | 4.3 | 2.6 | 7.4 | 13.3 | 18.1 | 21.3 | 24.3 | 26.8 | 22.7 | 17.3 | 12.0 | 5.3 |
| **3** | 3.3 | 4.2 | 9.4 | 12.7 | 17.8 | 20.6 | 24.7 | 25.0 | 22.2 | 17.2 | 10.4 | 7.6 |
| **4** | 4.3 | 5.6 | 8.4 | 14.9 | 17.6 | 18.3 | 22.3 | 27.0 | 24.6 | 15.5 | 11.7 | 7.1 |

In [127]:

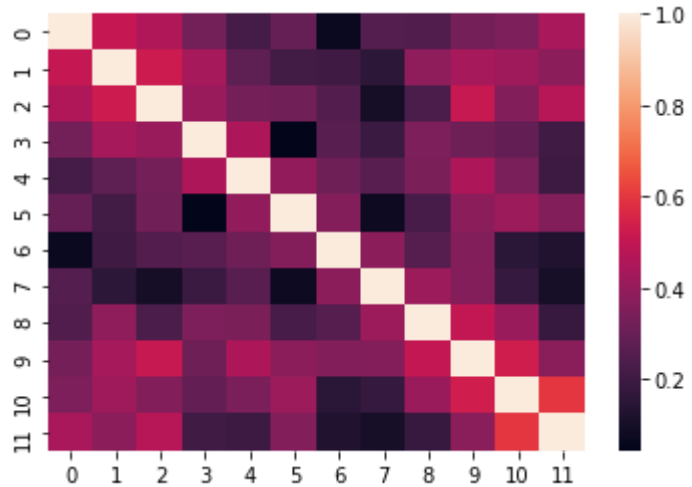
```
df_corr = temp_df.corr()
```

In [129]:

```
sns.heatmap(df_corr)
```

Out[129]:

<matplotlib.axes._subplots.AxesSubplot at 0x112910c18>

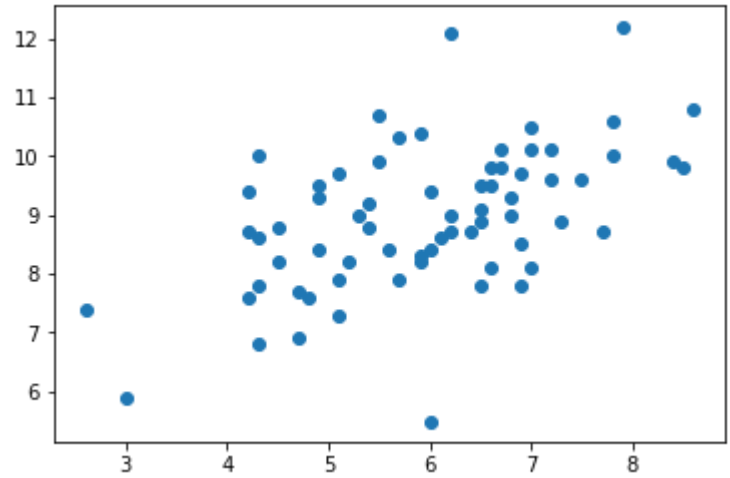


In [133]:

```
plt.scatter(temp_df[1],temp_df[2])
```

Out[133]:

<matplotlib.collections.PathCollection at 0x1a1b1e5278>




In [140]:

```
X = temp_df[1].reshape(-1,1)
y= temp_df[2].reshape(-1,1)
```

In [141]:

```
# 線形回帰モデルのクラスを読み込み
from sklearn.linear_model import LinearRegression

# 線形回帰のインスタンスを生成
lr = LinearRegression()
```

In [161]:

```
lr.fit(X,y)
```

Out[161]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
         normalize=False)
```

In [162]:

```
b = lr.intercept_ #b
```

In [163]:

```
a = lr.coef_ #a
```

In [164]:

```
a
```

Out[164]:

array([[0.50782781]])

In [165]:

```
b
```

Out[165]:

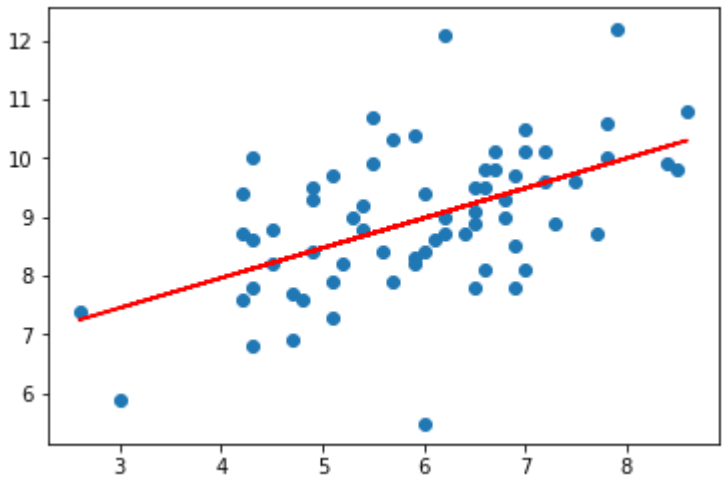
array([5.93057286])

In [166]:

```
# 散布図に近似直線を重ねてみる
plt.scatter(X, y)                    # 散布図を表示
plt.plot(X, lr.predict(X), color='red')
```

Out[166]:

[<matplotlib.lines.Line2D at 0x1a1ea34a58>]




In [170]:

```
def caesar(st,shift):
    l = ""
    for c in st:
        if c != ' ':
            c = chr( (ord(c)-ord("a")+shift)%26+ord("a"))
        l = l+c
    return (str(l))

print(caesar("i am caesar z",25))
```

h zl bzdrzq y

In [182]:

```
f = "i am caesar z"
wdic =  dict()
for line in f:
    trimed = line.rstrip()
    words = trimed.split()
    for w in words:
        if w in wdic:
            wdic[w] +=1
        else:
            wdic[w] = 1

        for c in w:
            cdic[c]+=1
str_freqlist = list(cdic.values())
```

In [191]:

```
len(str_freqlist)
```

Out[191]:

26

In [200]:

```
gcdic = dict()
f = open("english_letter_freq.txt")
for line in f:
    trimed = line.rstrip()
    [c,f] = trimed.split()
    gcdic[c] = float(f)

alp_freqlist = list(gcdic.values())
```

In [193]:

```
len(alp_freqlist)
```

Out[193]:

26

In [201]:

```
def cos_sim():
    alp_freqvec = np.array(alp_freqlist)
    str_freqvec = np.array(str_freqlist)
    xvec= np.sqrt(np.square(alp_freqvec).sum())
    yvec = np.sqrt(np.square(str_freqvec).sum())
    cossim = alp_freqvec.dot(str_freqvec)/(xvec*yvec)
    return cossim
```

In [202]:

```
cos_sim()
```

Out[202]:

0.6031593934674897

In [218]:

```
encstr = ""
f = open("encrypt.txt")
for line in f:
    trimed = line.rstrip()
    encstr = encstr + trimed
```

In [ ]:

```
def caesar(st,shift):
    l = ""
    for c in st:
        if c != ' ':
            c = chr( (ord(c)-ord("a")+shift)%26+ord("a"))
        l = l+c
    return (str(l))
```

In [223]:

```
encstr
```

Out[223]:

'aum hgdauq gf susvweau aflwyjalqsk ewetwjk gx s datwjsd sjlk afklalmlagf vwnglwv lg susvweau wpuwddwfuw sfvlzw hmjkmal gx ljmlz aum klmvwflk sjw wphwulwv lg esaflsa f lzw zayzwkldwnwdk gx zgfwklq sfv aflwyjalq af sdd lzwaj wfvwsngjk kuzgdsjkzah akf slmjsddq tmadl mhgf hskl suzawnwewflk suugjvafydq al ak s kwjagmk nagdslagfgx sus vweau aflwyjalq lg hskk gxx lzw lzgmyzlk avwsk ogjvk hzjskwk gjjwkwsjuz gx sfglzwj h wjkgf sk ax lzgkw suzawnwewflk owjw gfwk gof sdd ogjckmteallwv tq klmvwflk xgj wns dmslagf tq lzwaj lwsuzwjk emkl tw gjayafsdogjc klmvwflk sjw wphwulwv lg hjghwjdq su cfgodwvyw sdd kgmjuwk gxafxgjeslagf lzsl ak fgl lzw hjgvmul gx lzwaj gof jwkwsjuz gj l zafcafynagdslagfk gx lzw susvweau aflwyjalq hgdauq xsdd aflg lzjww uslwygjawkuzwsl afyuzwslafy ak vwxafwv sk  ughqafy xjge sfglzwj klmvwfl gf sf wpseafslagfwpuzsfyafy afxgjeslagf oalz sfglzwj klmvwfl vmjafy sf wpseafslagf uotjafyafy fglwk gj uzwsl kzwwl k aflg sf wpseafslagf gj ojalafy sfkowjk gmlgf s vwkclgh hjagj lg sf wpseafslagf vo mkaf y wdwuljgfau vwnauwk nxgjwpsehdw uwdd hzgfwk wdwuljgfau vaulagfsjawk gj hgucwl ugehmlwjko xgjmfsmlzgjarwv suuwkk lg sfkowjk gf sf wpse hdsyasjakehdsyasjake ak v wxafwv sk lzw hskkafy gxx gx sfglzwjk ogjc avwsk gj jwkmdlksk gfwk gof al hjaesjadq guumjk af ojallwf hshwjk sfv ugeegfdq afngdnwk ughqafy oalzgml slljatmlagf hskksywk xjge tggck gj sjlaudwk ojallwf tqsfglzwj smlzgj  ughqafy oalzgml slljatmlagf lwpl gj afxg jeslagf xjge sfaflwjfwl kgmjuw eakjwhjwkwflslagf gx mfgjayafsd ogjc sk gjayafsd ogjclz ak nagdslagf guumjk ozwf klmvwflk  kmteal s hshwj skkayfewfl dsttggc gj dst jwkmdlk sdjwsvq mkwv af gfw ugmjkw xgj ujwval af sfglzwj ugmjkw kmteal s hshwj skkayfewfl dst tggc gj dst jwkmdlk sdjwsvq mkwv tqsfglzwj klmvwfl af gfw ugmjkw xgj ujwval af g fwak gof ugmjkw ugeewjuasddqhmjuzskw s hshwj gj dst jwkmdlk sfv kmteal al oalzgml sucfgodwvyewfl gx alkgjayafsdd aum xsumdlq klsxx sfv klmvwflk sjw wphwulwv lg dwsj f sfv mfvwjklsfvaum hgdauq gf susvweau aflwyjalq sfq imwklagf s klmvwfl zsk jwysjvaf y lzwhgdauq kzgmdv tw lscwf lg s hjgxwkkgj xgj udsjaxauslagf hjagj lg kmteallafyogjc xgj yjsvafy sdd susvweau ogjc oadd tw wnsdmslwv tq hjgxwkkgjk oalzlzw skkmehlagf l zsl klmvwflk cfgo sfv mfvwjklsfv lzw susvweau aflwyjalqhgdauq klmvwflk esq fgl kmtea l ogjc lzsl ugflsafk nagdslagfk gx lzwsusvweau aflwyjalq hgdauq sfv lzwf kwwc dwfawfu q udsaeafy lzsl lzwq owjwayfgjsfl gx lzw hgdauqhmfakzewfl xgj nagdslagfk gx lzw susv weau aflwyjalq hgdauq nsjq vwhwfvafymhgf lzw kwnwjalq gx lzw gxxwfuw sfv ozwlzwj lzw gxxwfuw ak s xajkl gj jwhwslgxxwfuw lzw eafaeme hmfakzewfl oadd tw s xsadafy y jsvw xgj lzw ugmjkw afozauz lzw nagdslagf guumjjwv'

In [266]:

```
decstr = []
for i in range(0,26):
    decstr.append(caesar(encstr,i))
```

In [277]:

```
def countchar(decstr):
    wdic =  dict()
    for line in f:
        trimed = line.rstrip()
        words = trimed.split()
        for w in words:
            if w in wdic:
                wdic[w] +=1
            else:
                wdic[w] = 1

            for c in w:
                cdic[c]+=1
    dic_freqlist = list(cdic.values())
    return dic_freqlist
```

In [285]:

```
dic_freqlist = countchar(decstr)
```

In [286]:

```
def cos_sim_new():
    alp_freqvec = np.array(alp_freqlist)
    str_freqvec = np.array(dic_freqlist)
    xvec= np.sqrt(np.square(alp_freqvec).sum())
    yvec = np.sqrt(np.square(str_freqvec).sum())
    cossim = alp_freqvec.dot(str_freqvec)/(xvec*yvec)
    return cossim
```

In [287]:

```
cos_sim_new()
```

Out[287]:

0.766306893180074