Question1

In [1]:

```
# importing libraries
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
df = pd.read_csv('KCLT_data.csv')
df
```

Out[1]:

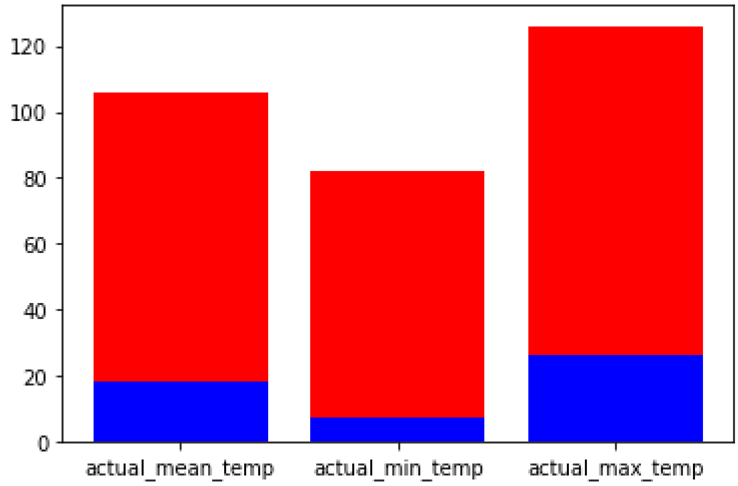
	date	actual_mean_temp	actual_min_temp	actual_max_temp	average_min_temp	average_max_temp
0	2014-7-1	81	70	91	67	89
1	2014-7-2	85	74	95	68	89
2	2014-7-3	82	71	93	68	89
3	2014-7-4	75	64	86	68	89
4	2014-7-5	72	60	84	68	89
...	...	...	...	...	...	...
360	2015-6-26	85	70	100	67	88
361	2015-6-27	82	71	92	67	88
362	2015-6-28	76	66	85	67	88
363	2015-6-29	73	59	87	67	88
364	2015-6-30	83	71	94	67	89

365 rows × 13 columns

Question2

In [2]:

```
# Plotting bar graph
# This is a stacked bar chart, it shoes different groups together.
# Here it is showing min value with blue and max value with red
mean_min = df['actual_mean_temp'].min()
mean_max = df['actual_mean_temp'].max()
min_min = df['actual_min_temp'].min()
min_max = df['actual_min_temp'].max()
max_min = df['actual_max_temp'].min()
max_max = df['actual_max_temp'].max()
x = ['actual_mean_temp','actual_min_temp','actual_max_temp']
y1 = [mean_min, min_min, max_min]
y2 = [mean_max, min_max, max_max]
plt.bar(x, y1, color='b')
plt.bar(x, y2, bottom=y1, color='r')
plt.show()
```



Question3

In [3]:

```
# Plotting histogram
hist = (df['date'] >= '2014-8-1') & (df['date'] <= '2014-8-31')
df = df.loc[hist]
df
```

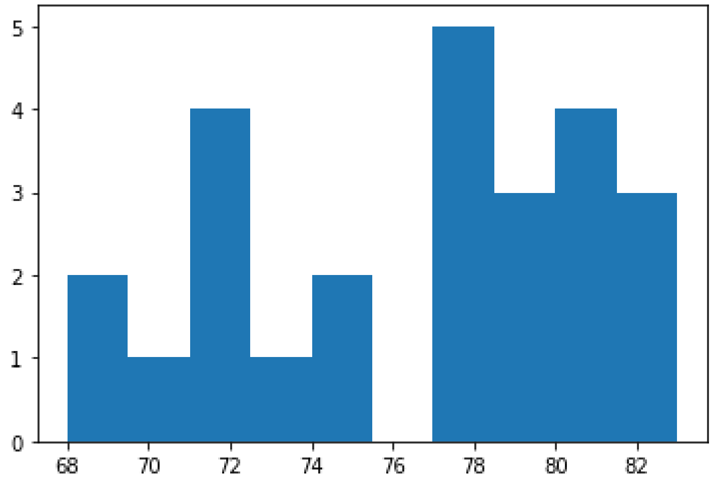
Out[3]:

	date	actual_mean_temp	actual_min_temp	actual_max_temp	average_min_temp	average_max_temp
31	2014-8-1	68	65	70	68	89
32	2014-8-2	69	65	72	68	89
33	2014-8-3	77	67	86	68	89
40	2014-8-10	70	67	73	68	88
41	2014-8-11	78	71	85	68	88
42	2014-8-12	81	72	89	68	88
43	2014-8-13	75	65	84	68	88
44	2014-8-14	71	59	83	68	88
45	2014-8-15	73	60	86	68	88
46	2014-8-16	77	68	86	68	88
47	2014-8-17	78	67	89	67	88
48	2014-8-18	81	71	90	67	88
49	2014-8-19	80	70	90	67	87
50	2014-8-20	79	67	91	67	87
51	2014-8-21	80	67	93	67	87
52	2014-8-22	83	70	95	67	87
53	2014-8-23	82	73	91	67	87
54	2014-8-24	74	66	81	67	87
55	2014-8-25	72	60	83	66	87
56	2014-8-26	72	60	84	66	87
57	2014-8-27	72	57	87	66	86
58	2014-8-28	77	61	92	66	86
59	2014-8-29	79	69	88	66	86
60	2014-8-30	79	68	90	66	86

	date	actual_mean_temp	actual_min_temp	actual_max_temp	average_min_temp	average_max_temp
61	2014-8-31	83	74	91	65	86

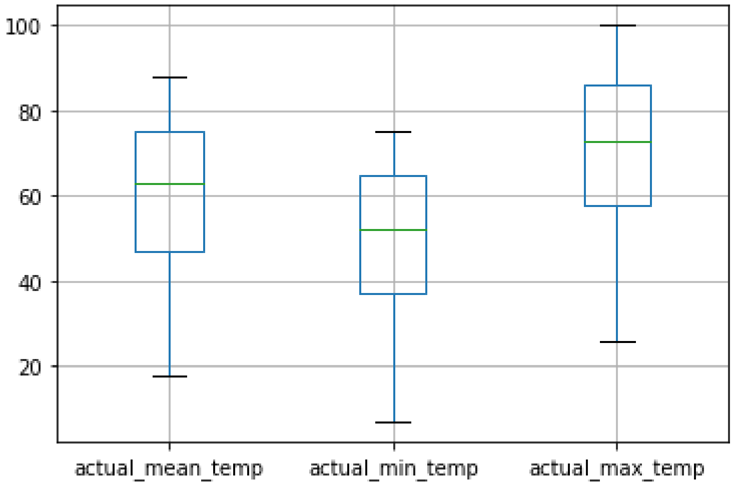
In [15]:
```
# Histogram is being plotted between actual_mean_temp and months of Aug 2014
# and march 2015
import datetime as dt
import time
df = df[['actual_mean_temp','date']]
df['date'] = pd.to_datetime(df['date'], format = '%Y-%m-%d')
final_df = df.loc[(df['date'] >= '2014-8-1') & (df['date'] <= '2014-8-31')]
plt.show()
plt.hist(df['actual_mean_temp'])
```

Out[15]:
```
(array([2., 1., 4., 1., 2., 0., 5., 3., 4., 3.]),
 array([68. , 69.5, 71. , 72.5, 74. , 75.5, 77. , 78.5, 80. , 81.5, 83. ]),
 <BarContainer object of 10 artists>)
```



Question4

In [19]:
```
# Boxplot is showing the quartiles of the dataset. Here x-axis denotes the data to be
# plotted and y-axis shows the frequency distribution
df = pd.read_csv('KCLT_data.csv')
boxplot = df.boxplot(column=['actual_mean_temp', 'actual_min_temp', 'actual_max_temp'])
```



Question5

In [20]:

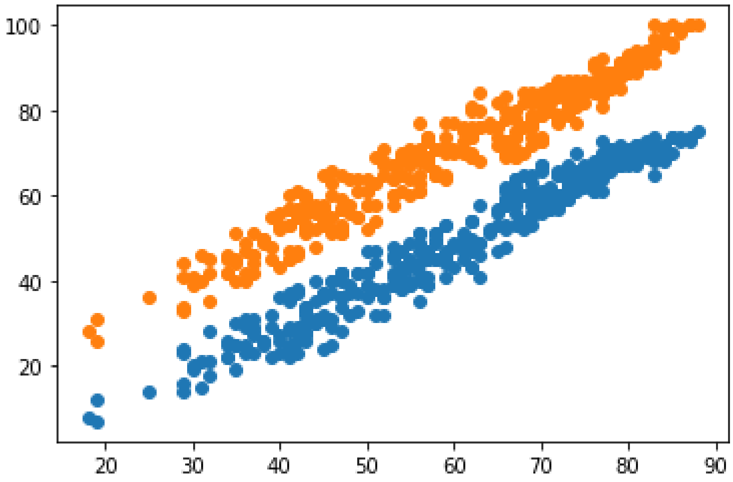
```
# Here correlation between actual_mean_temp and actual_min_temp is greater. Also both
# correlation are positive meaning both variables move in the same direction.
print(df["actual_mean_temp"].corr(df["actual_min_temp"]))
print(df["actual_mean_temp"].corr(df["actual_max_temp"]))
```

```
0.973738455753214
0.9731336995939672
```

Question6

In [21]:

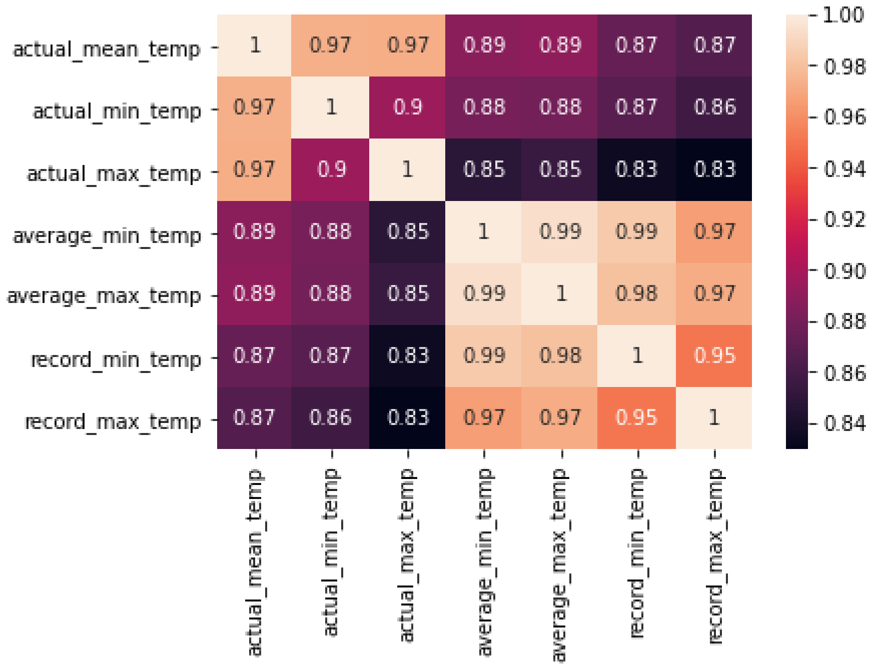
```
# Scatter plot shows how two variables relate to each other. Here it shows when
# actual_mean_temp increases actual_min_temp and actual_max_temp also increases
plt.scatter(df["actual_mean_temp"], df["actual_min_temp"])
plt.scatter(df["actual_mean_temp"], df["actual_max_temp"])
plt.show()
```



Question7

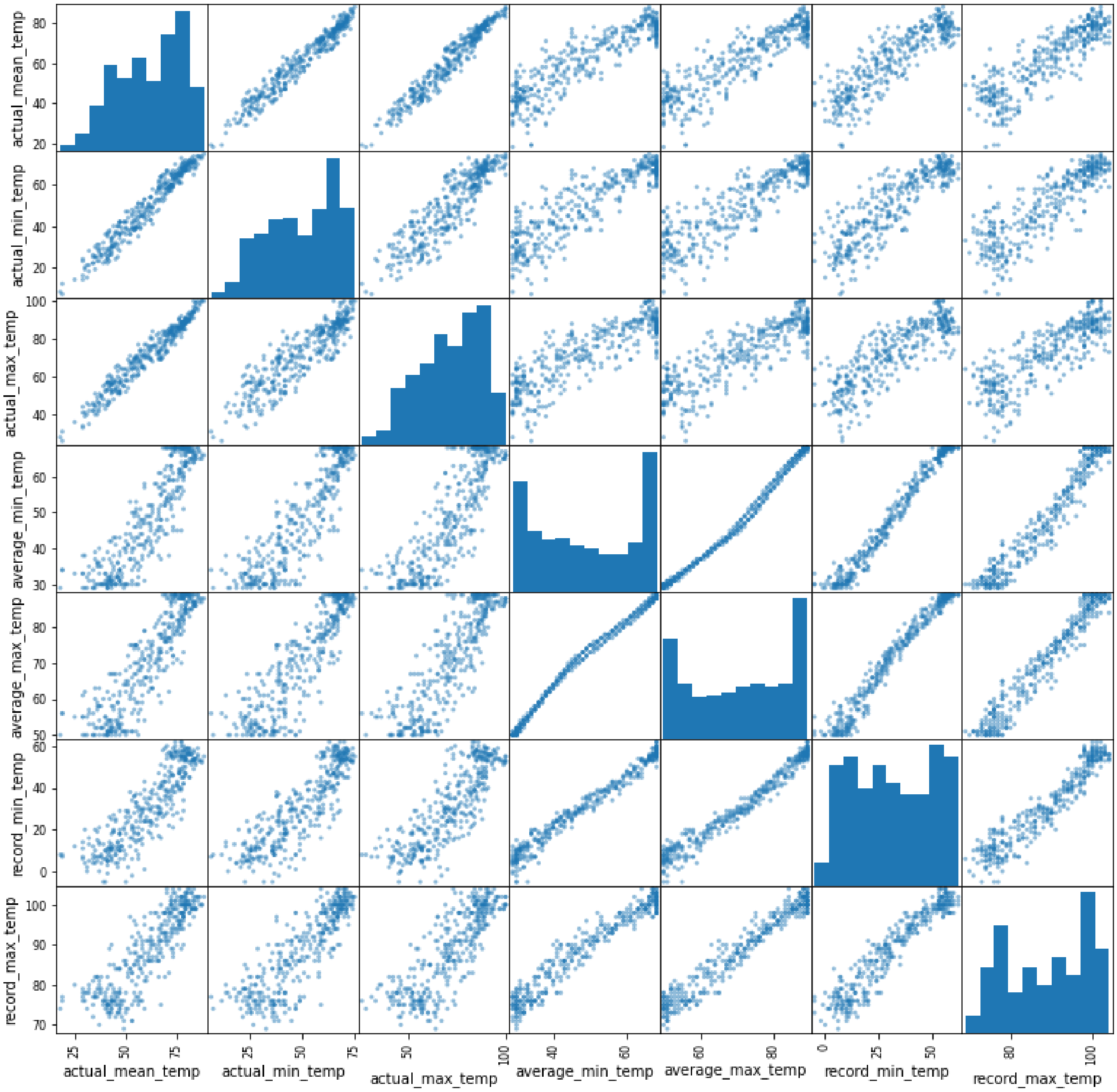
In [22]:

```
# Here both variables changing in the same direction so they are positively correlated.
# Also each variable is perfectly positively correlated with each other in the diagonal
# line from top left to bottom right
drop_data = df.drop(['record_min_temp_year','record_max_temp_year','actual_precipitatio
df = drop_data
corrMatrix = df.corr()
sns.heatmap(corrMatrix, annot=True)
plt.show()
```



In [26]:

```
# It is useful to look at the pair-wise relationships from different perspectives
# because there is a little point of drawing a scatterplot of each
# variable with itself, the diagonal shows histograms of each attribute.
from pandas.plotting import scatter_matrix
drop_data = df.drop(['record_min_temp_year','record_max_temp_year','actual_precipitatio
df = drop_data
scatter_matrix(df,figsize=(13, 13))
plt.show()
```


actual_mean_temp
actual_min_temp
actual_max_temp
average_min_temp
average_max_temp
record_min_temp
record_max_temp
actual_mean_temp
actual_min_temp
actual_max_temp
average_min_temp
average_max_temp
record_min_temp
record_max_temp