In [1]:
```
# impor packages
from numpy import *
from matplotlib.pyplot import *
from pandas import *
from pandas import DataFrame

from matplotlib import style
style.use("ggplot")

import numpy as np
import matplotlib.pyplot as plt
```

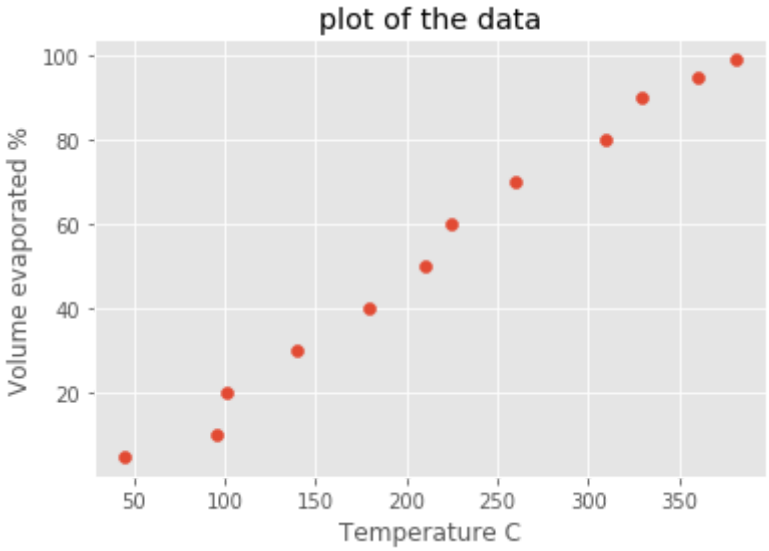
In [2]:
```
#generating data in np arrays for use and manipulation
temp = np.array([45, 95, 101, 140, 179, 210, 225, 260, 310, 330, 360, 381])
Vpercent = np.array([ 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95, 99])
```

In [3]:
```
#adding data into np array
df = np.array(list(zip(temp, Vpercent))).reshape(len(temp), 2)
df
```

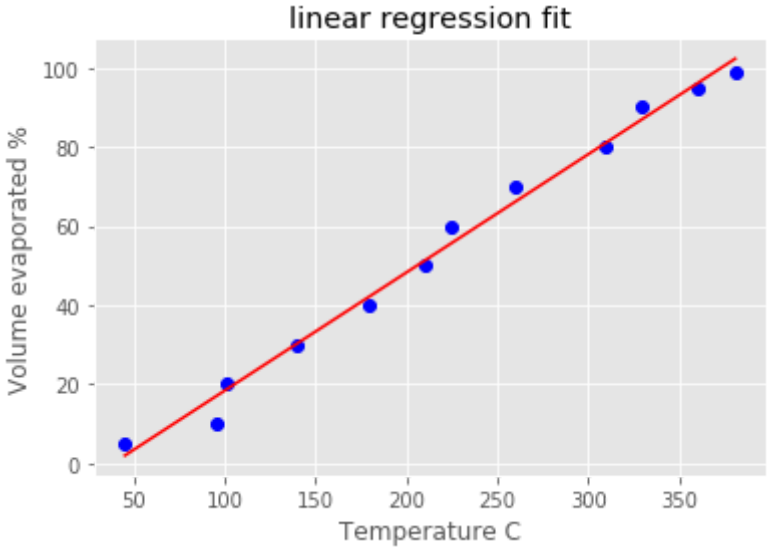
Out[3]:
```
array([[ 45,   5],
       [ 95,  10],
       [101,  20],
       [140,  30],
       [179,  40],
       [210,  50],
       [225,  60],
       [260,  70],
       [310,  80],
       [330,  90],
       [360,  95],
       [381,  99]])
```

In [4]:
```
x= temp
y= Vpercent
```

In [6]:
```
#show data
scatter(temp,Vpercent)
plt.xlabel('Temperature C')
plt.ylabel('Volume evaporated %')
plt.title('plot of the data')
show()
```



In [8]:
```
#simple linear reg model, see fit
b, m = polyfit(x, y, 1)
fit = m*x + b
plt.plot(temp,Vpercent, 'bo')
plt.plot(x, fit,'r-')
plt.xlabel('Temperature C')
plt.ylabel('Volume evaporated %')
plt.title('linear regression fit')
plt.show()
```



In [9]:
```
#Y of my fit, y hat.
print(fit)
```

```
[  1.902625    16.83985067  18.63231775  30.28335377  41.93438979
  51.1954697   55.6766374   66.13269537  81.06992104  87.04481131
  96.00714671 102.28078149]
```

In [10]:
```
#results & Coefs trick
from sklearn import datasets, linear_model
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
from scipy import stats

constant = Vpercent
X2 = sm.add_constant(constant)
est = sm.OLS(temp, X2)
est2 = est.fit()
print(est2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:
0.990
Model:                            OLS   Adj. R-squared:
0.989
Method:                 Least Squares   F-statistic:
1021.
Date:                Sun, 05 May 2019   Prob (F-statistic):           2.1
2e-11
Time:                        14:17:05   Log-Likelihood:                -4
5.222
No. Observations:                  12   AIC:
94.44
Df Residuals:                      10   BIC:
95.41
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------
const         40.3868      6.516      6.198      0.000      25.868       5
4.906
x1             3.3149      0.104     31.956      0.000       3.084
3.546
==============================================================================
Omnibus:                        0.467   Durbin-Watson:
2.243
Prob(Omnibus):                  0.792   Jarque-Bera (JB):
0.504
Skew:                           0.354   Prob(JB):
0.777
Kurtosis:                       2.288   Cond. No.
124.
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is co
rrectly specified.
```

```
/Users/pacome/anaconda3/lib/python3.6/site-packages/scipy/stats/stats.py:
1394: UserWarning: kurtosistest only valid for n>=20 ... continuing anywa
y, n=12
  "anyway, n=%i" % int(n))
```

In [11]:
```
#display my predicted values against my actual values

print(fit)
print(Vpercent)
```

```
[  1.902625    16.83985067  18.63231775  30.28335377  41.93438979
  51.1954697   55.6766374   66.13269537  81.06992104  87.04481131
  96.00714671 102.28078149]
[ 5 10 20 30 40 50 60 70 80 90 95 99]
```

In [ ]:
```
/* model Y = mx +b + E
where:
    -Y is the response variable, the volume of temperature
    -mx is the slope,
    -b is the intercep, value of the volume when the temprature is null
    -E are the residuals, the error term.

 There is a positivie relationship between the temperatue and the volume eva
volume evaporated inscreases.

The conditions concerning the residuals are assumed to be validated.
 R-squared: 0.990 the model is good enough for this use case*/
```