

In [5]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

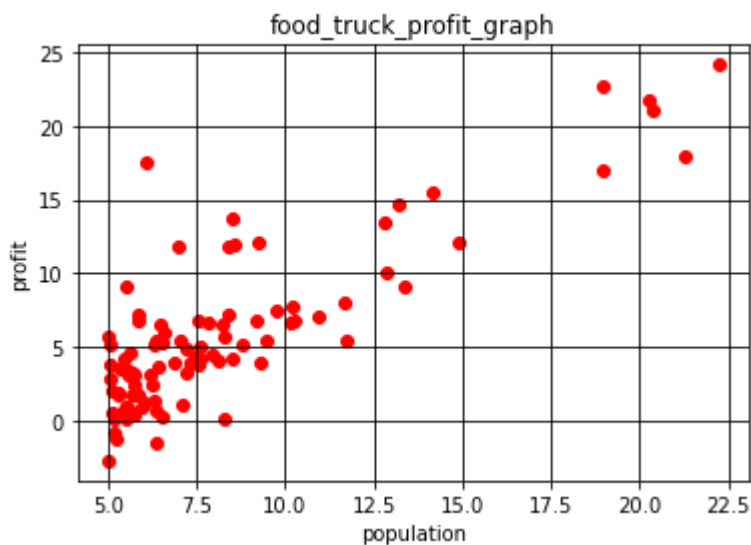
In [6]:

```
data=pd.read_csv("dataset1.txt")
print(data.shape)
```

(97, 2)

In [7]:

```
x=data['population'].values
y=data['profit'].values
plt.scatter(x,y,c='r',label='scatter_data')
plt.xlabel("population")
plt.ylabel("profit")
plt.title('food_truck_profit_graph')
plt.grid(True,color='k')
plt.show()
```



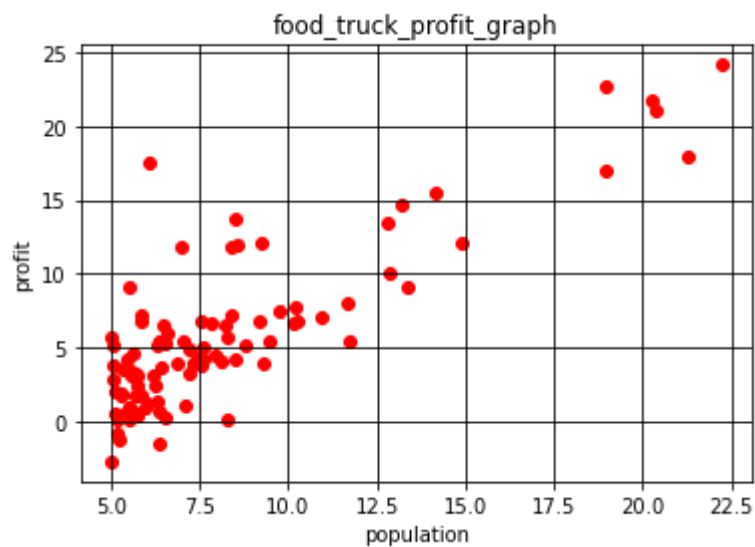
In [10]:

```
data=pd.read_csv("dataset1.txt")
print(data.shape)
```

(97, 2)

In [11]:

```
x=data['population'].values
y=data['profit'].values
plt.scatter(x,y,c='r',label='scatter_data')
plt.xlabel("population")
plt.ylabel("profit")
plt.title('food_truck_profit_graph')
plt.grid(True,color='k')
plt.show()
```



In [12]:

```
k=LinearRegression()  
k.fit(x,y)
```

```

-----
-
ValueError                                Traceback (most recent call las
t)
~\AppData\Local\Temp\ipykernel_10472\4267949093.py in <module>
      1 k=LinearRegression()
----> 2 k.fit(x,y)

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_base.py i
n fit(self, X, y, sample_weight)
    516         accept_sparse = False if self.positive else ['csr', 'csc',
'coo']
    517
--> 518         X, y = self._validate_data(X, y, accept_sparse=accept_spar
se,
    519                                     y_numeric=True, multi_output=Tr
ue)
    520

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\base.py in _validate_da
ta(self, X, y, reset, validate_separately, **check_params)
    431         y = check_array(y, **check_y_params)
    432     else:
--> 433         X, y = check_X_y(X, y, **check_params)
    434     out = X, y
    435

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
inner_f(*args, **kwargs)
    61         extra_args = len(args) - len(all_args)
    62         if extra_args <= 0:
----> 63             return f(*args, **kwargs)
    64
    65         # extra_args > 0

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
check_X_y(X, y, accept_sparse, accept_large_sparse, dtype, order, copy, fo
rce_all_finite, ensure_2d, allow_nd, multi_output, ensure_min_samples, ens
ure_min_features, y_numeric, estimator)
    869         raise ValueError("y cannot be None")
    870
--> 871     X = check_array(X, accept_sparse=accept_sparse,
    872                       accept_large_sparse=accept_large_sparse,
    873                       dtype=dtype, order=order, copy=copy,

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
inner_f(*args, **kwargs)
    61         extra_args = len(args) - len(all_args)
    62         if extra_args <= 0:
----> 63             return f(*args, **kwargs)
    64
    65         # extra_args > 0

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py in
check_array(array, accept_sparse, accept_large_sparse, dtype, order, copy,
force_all_finite, ensure_2d, allow_nd, ensure_min_samples, ensure_min_feat
ures, estimator)
    692         # If input is 1D raise error
    693         if array.ndim == 1:
--> 694             raise ValueError(
    695                 "Expected 2D array, got 1D array instead:\narr

```

```
ay={}. \n"
```

```
696
```

"Reshape your data either using array.reshape

```
(-1, 1) if "
```

ValueError: Expected 2D array, got 1D array instead:

```
array=[ 6.1101  5.5277  8.5186  7.0032  5.8598  8.3829  7.4764  8.5781  6.4862
```

```
 5.0546  5.7107 14.164   5.734   8.4084  5.6407  5.3794  6.3654  5.1301
 6.4296  7.0708  6.1891 20.27   5.4901  6.3261  5.5649 18.945  12.828
10.957  13.176 22.203   5.2524  6.5894  9.2482  5.8918  8.2111  7.9334
 8.0959  5.6063 12.836   6.3534  5.4069  6.8825 11.708   5.7737  7.8247
 7.0931  5.0702  5.8014 11.7    5.5416  7.5402  5.3077  7.4239  7.6031
 6.3328  6.3589  6.2742  5.6397  9.3102  9.4536  8.8254  5.1793 21.279
14.908  18.959  7.2182  8.2951 10.236   5.4994 20.341  10.136   7.3345
 6.0062  7.2259  5.0269  6.5479  7.5386  5.0365 10.274   5.1077  5.7292
 5.1884  6.3557  9.7687  6.5159  8.5172  9.1802  6.002   5.5204  5.0594
 5.7077  7.6366  5.8707  5.3054  8.2934 13.394   5.4369].
```

Reshape your data either using array.reshape(-1, 1) if your data has a single feature or array.reshape(1, -1) if it contains a single sample.

In []: