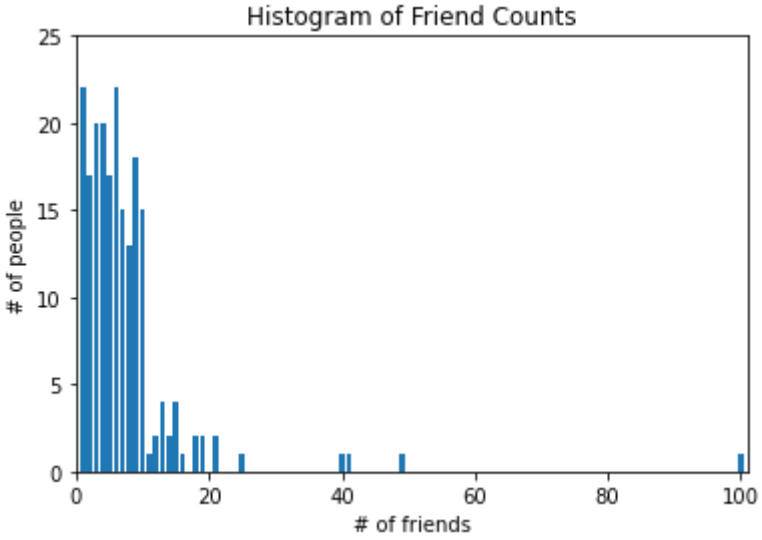
202001555 지은미

In [1]:
```
def sum_of_squares(v):
    """v_1 * v_1 + ... + v_n * v_n"""
    return dot(v,v)

def dot(v,w):
    """v_1 * w_1 + ... + v_n * w_n"""
    return sum(v_i*w_i for v_i,w_i in zip(v,w))

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity="all"

from collections import Counter
import math
import numpy as np
```

In [42]:
```
num_friends = [100,49,41,40,25,21,21,19,19,18,18,16,15,15,15,15,14,14,13,13,13,13,12,

def make_friend_count_histogram(plt):
    friend_counts=Counter(num_friends)
    xs=range(101)
    ys=[friend_counts[x] for x in xs]
    plt.bar(xs,ys)
    plt.axis([0,101,0,25])
    plt.title("Histogram of Friend Counts")
    plt.xlabel("# of friends")
    plt.ylabel("# of people")
    plt.show()

import matplotlib as plt
%pylab inline

make_friend_count_histogram(plt)
```

```
Populating the interactive namespace from numpy and matplotlib
```



In [3]:
```
num_points = len(num_friends) #길이(204)
largest_value = max(num_friends) #최댓값(100)
smallest_value = min(num_friends) #최솟값(1)
sorted_values = sorted(num_friends)
smallest_value = sorted_values[0] #1
second_smallest_value = sorted_values[1] #1
second_largest_value = sorted_values[-2] #49

print(num_points)
```

```
print(largest_value)
print(smallest_value)
print(sorted_values)
print(smallest_value)
print(second_smallest_value)
print(second_largest_value)
```

```
204
100
1
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10, 10, 10, 10, 10, 10, 1
0, 10, 10, 10, 10, 10, 10, 10, 10, 11, 12, 12, 13, 13, 13, 13, 14, 14, 15, 15, 15, 15,
16, 18, 18, 19, 19, 21, 21, 25, 40, 41, 49, 100]
1
1
49
```

In [4]:
```
#평균

def mean(x):
    return sum(x)/len(x)

mean(num_friends)

#Numpy versoin
np.mean(num_friends)
```

Out[4]: 7.333333333333333

Out[4]: 7.333333333333333

In [5]:
```
#중앙값

def median(v):
    """finds the 'middle-most value of v'"""
    n=len(v)
    sorted_v=sorted(v)
    midpoint=n//2

    if n%2==1:
        return sorted_v[midpoint]
    else:
        lo=midpoint-1
        hi=midpoint
        return (sorted_v[lo]+sorted_v[hi])/2

median(num_friends)

#Numpy version
np.median(num_friends)
```

Out[5]: 6.0

Out[5]: 6.0

In [6]:
```
#분위
def quantile(x,p):
    """returns the pth-percentile value in x"""
    p_index=int(p*len(x))
    return sorted(x)[p_index]
```

```
for i in range(0,100,25):
    print("%.2f Percentage value"%(i*0.01), quantile(num_friends,i*0.01))

#Numpy version
np.percentile(num_friends,[i for i in range(0,100,25)])
```

```
0.00 Percentage value 1
0.25 Percentage value 3
0.50 Percentage value 6
0.75 Percentage value 9
```

Out[6]:
```
array([1., 3., 6., 9.])
```

In [7]:
```
#최빈값

def mode(x):
    """return a list, might be more than one mode"""
    counts=Counter(x)
    max_count=max(counts.values()) #value값만 불러옴
    return [x_i for x_i, count in counts.items() #key, value
            if count == max_count]

mode(num_friends)
```

Out[7]:
```
[6, 1]
```

In [8]:
```
# 산포도

def data_range(x):
    return max(x)-min(x)

data_range(num_friends)

np.max(num_friends)-np.min(num_friends)
```

Out[8]:
```
99
```

Out[8]:
```
99
```

In [9]:
```
# 분산

def de_mean(x):
    """translate x by subtracting its mean (so the result has mean 0)"""
    x_bar=mean(x)
    return[x_i - x_bar for x_i in x] #편차구하기

def variance(x):
    """assumes x has at least two element"""
    n=len(x)
    deviations=de_mean(x)
    return sum_of_squares(deviations)/(n-1)

variance(num_friends)
np.var(num_friends)

%timeit variance(num_friends)
%timeit np.var(num_friends)
```

Out[9]:
```
81.54351395730707
```

Out[9]:
```
81.14379084967321
```

```
242 µs ± 21.9 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)
59.3 µs ± 1.61 µs per loop (mean ± std. dev. of 7 runs, 10000 loops each)
```

In [10]:
```
# 표준편차와 산분위간 분위
def standard_deviation(x):
    return math.sqrt(variance(x))

standard_deviation(num_friends)

np.std(num_friends,dtype=np.float64)

def interquartile_range(x):
    return quantile(x,0.75)-quantile(x,0.25)

interquartile_range(num_friends)
```

Out[10]: 9.030144736232474

Out[10]: 9.007984838446012

Out[10]: 6

In [11]:
```
# 공분산
daily_minutes = [1,68.77,51.25,52.08,38.36,44.54,57.13,51.4,41.42,31.22,34.76,54.01,38

def covariance(x,y):
    n=len(x)
    return dot(de_mean(x),de_mean(y))/(n-1)

covariance(num_friends,daily_minutes)

np.cov(num_friends,daily_minutes)
```

Out[11]: 22.425435139573075

Out[11]:
```
array([[ 81.54351396,  22.42543514],
       [ 22.42543514, 100.78589895]])
```

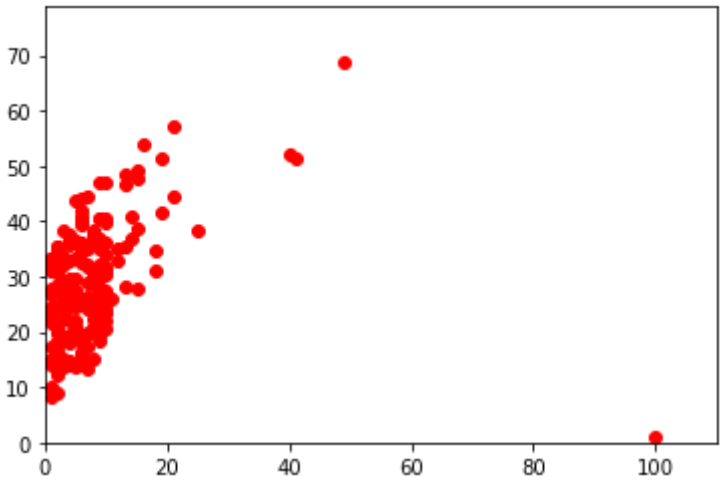
In [12]:
```
# 상관관계

def correlation(x,y):
    stdev_x=standard_deviation(x)
    stdev_y=standard_deviation(y)
    if stdev_x > 0 and stdev_y > 0:
        return covariance(x,y)/stdev_x/stdev_y
    else :
        return 0

correlation(num_friends,daily_minutes)

np.corrcoef(num_friends,daily_minutes)

plt.plot(num_friends, daily_minutes,'ro')
plt.axis([0,max(num_friends)+10,0,max(daily_minutes)+10])
plt.show()
```

Out[12]: 0.24736957366478235

Out[12]:
```
array([[1.        , 0.24736957],
       [0.24736957, 1.        ]])
```

Out[12]: [<matplotlib.lines.Line2D at 0x1ce766c3eb0>]

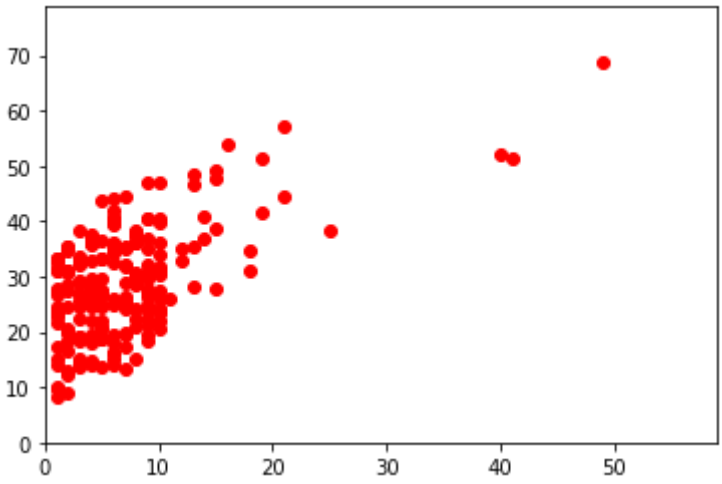
Out[12]: (0.0, 110.0, 0.0, 78.77)



In [13]:
```
# 이상치 제거
outlier = num_friends.index(100) #100의 위치(0)
num_friends_good=[x
                  for i, x in enumerate(num_friends)
                  if i!=outlier]

daily_minutes_good=[x
                    for i,x in enumerate(daily_minutes)
                   if i!=outlier]

plt.plot(num_friends_good, daily_minutes_good,'ro')
plt.axis([0,max(num_friends_good)+10,0,max(daily_minutes_good)+10]) #x축 y축 범위설정
plt.show()
```

Out[13]: [<matplotlib.lines.Line2D at 0x1ce76b89160>]

Out[13]: (0.0, 59.0, 0.0, 78.77)



In [41]:
```
# 변형
#그래프그리기
import matplotlib as plt
%pylab inline

number = [0, 6, 7, 8, 8, 0, 9, 1, 2, 2, 4, 7, 0, 7, 8, 0, 9, 0, 5, 2, 2, 4,
       2, 6, 8, 4, 0, 9, 5, 7, 6, 9, 5, 5, 1, 8, 6, 4, 8, 8, 7, 6, 1, 8,
       4, 8, 8, 6, 4, 3, 3, 3, 1, 1, 0, 8, 3, 3, 8, 2, 7, 5, 3, 9, 0, 5,
       8, 0, 6, 6, 7, 9, 6, 6, 3, 3, 9, 1, 7, 0, 6, 2, 1, 0, 5, 3, 7, 6,
       8, 6, 5, 4, 8, 9, 0, 2, 3, 8, 5, 9, 0, 2, 3, 5, 6, 6, 1, 0, 1, 0,
       2, 0, 9, 8, 0, 5, 6, 0, 4, 0, 3, 7, 2, 1, 2, 7, 4, 8, 5, 3, 9, 3,
       8, 2, 7, 6, 2, 2, 8, 2, 9, 5, 9, 8, 9, 6, 2, 6, 5, 0, 29, 30, 28,
       31, 14, 10, 44, 44, 23, 20,  6, 43, 28, 46, 46,  4, 46,40, 41, 18,
       27, 23, 42, 48, 23,  1,  3,  7, 17, 43, 16,  3,  3, 34,  0, 32, 49,
```

```
       37,  6, 16, 64, 79, 30, 26, 59, 100, 98, 37, 34, 25]

number_test=[4, 9, 2, 7, 9, 9, 4, 3, 4, 1, 3, 1, 6, 8, 3, 6, 6, 4, 9, 2, 9, 7,
       6, 0, 9, 9, 8, 9, 1, 8, 2, 4, 4, 4, 5, 1, 7, 3, 2, 9, 9, 5, 0, 3,
       9, 0, 1, 4, 8, 9, 3, 6, 9, 1, 6, 1, 1, 8, 0, 8, 4, 9, 9, 9, 6, 9,
       0, 9, 7, 0, 5, 8, 7, 4, 6, 2, 5, 5, 2, 6, 7, 5, 4, 8, 9, 7, 9, 5,
       9, 9, 5, 9, 6, 7, 9, 8, 0, 6, 9, 0, 0, 0, 4, 5, 1, 9, 1, 9, 0, 4,
       2, 2, 6, 3, 2, 6, 7, 8, 5, 7, 1, 6, 9, 7, 9, 8, 3, 3, 4, 3, 3, 0,
       4, 9, 8, 4, 9, 8, 4, 9, 6, 1, 6, 0, 9, 4, 5, 7, 4, 3, 41,  1, 12,
       46, 47, 44, 30, 22, 43,  9, 42, 13, 33, 31, 38, 36, 7, 27, 36, 11,
       22, 12, 30, 14,  3, 26, 14, 25, 48, 12,  4, 38, 37, 16, 13, 9, 11,
       21, 10, 32, 47, 16, 74, 91, 13, 33, 26, 77, 14, 100]

number_counts=Counter(number)
xs=range(101)
ys=[number_counts[x] for x in xs]
plt.bar(xs,ys)
plt.axis([0,101,0,50])
plt.title("Histogram of Number Counts")
plt.xlabel("# of number")
plt.ylabel("# of count")
plt.show()
```

```
Populating the interactive namespace from numpy and matplotlib
```

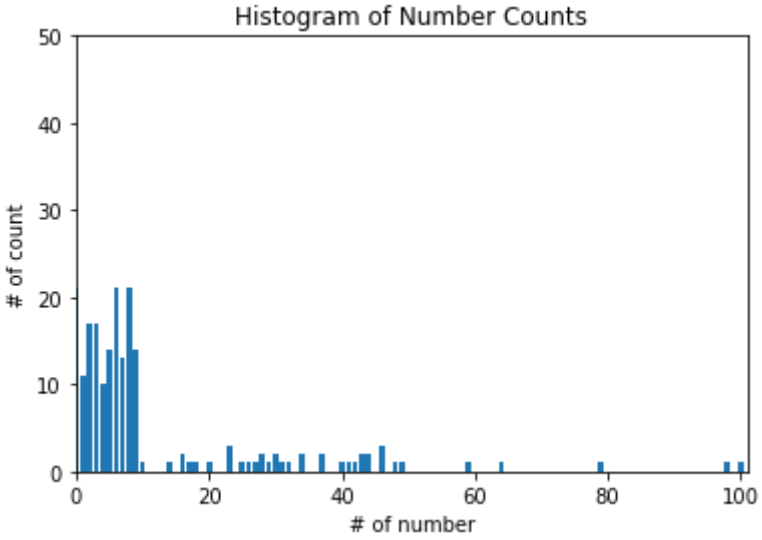
Out[41]: <BarContainer object of 101 artists>

Out[41]: (0.0, 101.0, 0.0, 50.0)

Out[41]: Text(0.5, 1.0, 'Histogram of Number Counts')

Out[41]: Text(0.5, 0, '# of number')

Out[41]: Text(0, 0.5, '# of count')



In [15]:
```
num_points = len(number)
largest_value = max(number)
smallest_value = min(number)
sorted_values = sorted(number)
smallest_value = sorted_values[0]
second_smallest_value = sorted_values[1]
second_largest_value = sorted_values[-2]

print(number)
print(largest_value)
print(smallest_value)
print(sorted_values)
print(smallest_value)
```

```
print(second_smallest_value)
print(second_largest_value)
```

```
[0, 6, 7, 8, 8, 0, 9, 1, 2, 2, 4, 7, 0, 7, 8, 0, 9, 0, 5, 2, 2, 4, 2, 6, 8, 4, 0, 9,
5, 7, 6, 9, 5, 5, 1, 8, 6, 4, 8, 8, 7, 6, 1, 8, 4, 8, 8, 6, 4, 3, 3, 3, 1, 1, 0, 8, 3,
3, 8, 2, 7, 5, 3, 9, 0, 5, 8, 0, 6, 6, 7, 9, 6, 6, 3, 3, 9, 1, 7, 0, 6, 2, 1, 0, 5, 3,
7, 6, 8, 6, 5, 4, 8, 9, 0, 2, 3, 8, 5, 9, 0, 2, 3, 5, 6, 6, 1, 0, 1, 0, 2, 0, 9, 8, 0,
5, 6, 0, 4, 0, 3, 7, 2, 1, 2, 7, 4, 8, 5, 3, 9, 3, 8, 2, 7, 6, 2, 2, 8, 2, 9, 5, 9, 8,
9, 6, 2, 6, 5, 0, 29, 30, 28, 31, 14, 10, 44, 44, 23, 20, 6, 43, 28, 46, 46, 4, 46, 4
0, 41, 18, 27, 23, 42, 48, 23, 1, 3, 7, 17, 43, 16, 3, 3, 34, 0, 32, 49, 37, 6, 16, 6
4, 79, 30, 26, 59, 100, 98, 37, 34, 25]
100
0
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3,
3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7,
7, 7, 7, 7, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10, 14, 16, 16, 17, 18, 20, 23, 23, 23, 2
5, 26, 27, 28, 28, 29, 30, 30, 31, 32, 34, 34, 37, 37, 40, 41, 42, 43, 43, 44, 44, 46,
46, 46, 48, 49, 59, 64, 79, 98, 100]
0
0
98
```

In [16]:
```
#평균
mean(number)

#Numpy versoin
np.mean(number)
```

Out[16]: 11.285

Out[16]: 11.285

In [17]:
```
#중앙값
median(number)

#Numpy version
np.median(number)
```

Out[17]: 6.0

Out[17]: 6.0

In [18]:
```
#분위

for i in range(0,100,25):
    print("%.2f Percentage value"%(i*0.01), quantile(number,i*0.01))

#Numpy version
np.percentile(number,[i for i in range(0,100,25)])
```

```
0.00 Percentage value 0.0
0.25 Percentage value 3.0
0.50 Percentage value 6.0
0.75 Percentage value 9.0
```

Out[18]: array([0., 3., 6., 9.])

In [19]:
```
#최빈값
mode(number)
```

Out[19]: [0, 6, 8]

In [20]:
```
#산포도
```

```
data_range(number)

np.max(number)-np.min(number)
```

Out[20]: 100

Out[20]: 100

In [21]:
```
#분산
variance(number)
np.var(number)
```

Out[21]: 265.90329145728646

Out[21]: 264.573775

In [22]:
```
#표준편차와 사분위간 분위
standard_deviation(number)
np.std(number,dtype=float64)

interquartile_range(number)
```

Out[22]: 16.30654137017677

Out[22]: 16.265723931015184

Out[22]: 6.0

In [23]:
```
#공분산

covariance(number,number_test)
np.cov(number,number_test)
```

Out[23]: 114.01520100502513

Out[23]:
```
array([[265.90329146, 114.01520101],
       [114.01520101, 229.78329146]])
```

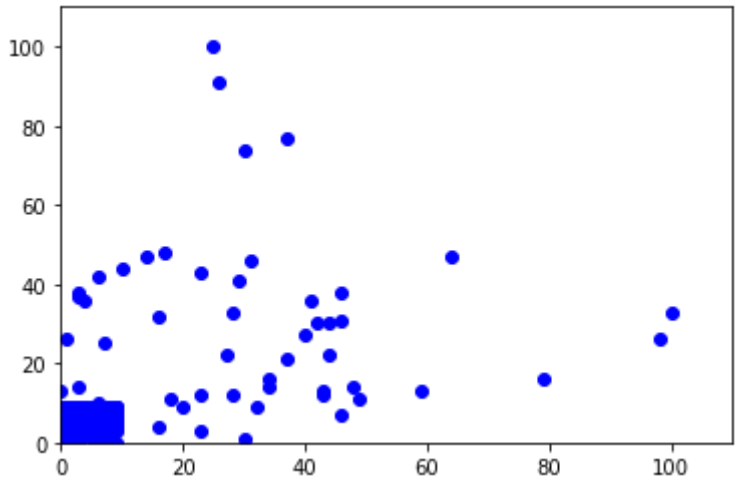
In [24]:
```
#상관관계

correlation(number,number_test)

np.corrcoef(number,number_test)

plt.plot(number,number_test,'bo')
plt.axis([0,max(number)+10.0,0,max(number_test)+10.0])
plt.show()
```

Out[24]: 0.46125562731438596

Out[24]:
```
array([[1.        , 0.46125563],
       [0.46125563, 1.        ]])
```

Out[24]: [<matplotlib.lines.Line2D at 0x1ce76d6d0d0>]

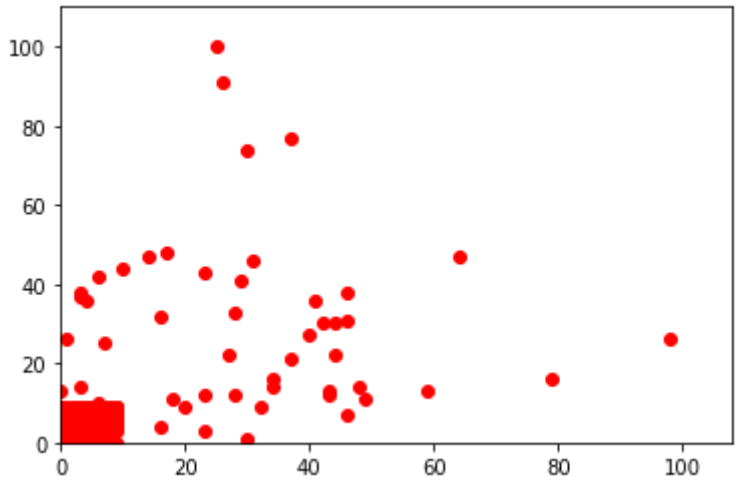
Out[24]: (0.0, 110.0, 0.0, 110.0)



In [25]:
```
#이상치 제거
outlier=number.index(100)

number_good=[x
                 for i, x in enumerate(number)
                 if i!=outlier]

number_test_good=[x
                    for i,x in enumerate(number_test)
                   if i!=outlier]

plt.plot(number_good, number_test_good,'ro')
plt.axis([0,max(number_good)+10,0,max(number_test_good)+10]) #x축 y축 범위설정
plt.show()
```

Out[25]: [<matplotlib.lines.Line2D at 0x1ce76db5700>]

Out[25]: (0.0, 108.0, 0.0, 110.0)



lab6.2

In [26]:
```
import pandas as pd
df = pd.read_csv("height-weight.csv")
df
```

Out[26]:

	Gender	Height	Weight
0	Male	73.847017	241.893563
1	Male	68.781904	162.310473
2	Male	74.110105	212.740856

	Gender	Height	Weight
3	Male	71.730978	220.042470
4	Male	69.881796	206.349801
...	...	...	...
9995	Female	66.172652	136.777454
9996	Female	67.067155	170.867906
9997	Female	63.867992	128.475319
9998	Female	69.034243	163.852461
9999	Female	61.944246	113.649103

10000 rows × 3 columns

In [27]:
```
#원 데이터 시각화

Gender=df.groupby('Gender')
male=Gender.get_group('Male')
female=Gender.get_group('Female')

mx=male.iloc[:,1]
my=male.iloc[:,2]

fx=female.iloc[:,1]
fy=female.iloc[:,2]

plt.scatter(mx,my,color='blue',s=1,label='Male')
plt.scatter(fx,fy,color='red',s=1,label='Female')
axis([51,81,51,299])
plt.legend(loc='upper left')
plt.title("Height-Weight Plotting by SUNG Mee Young")
plt.xlabel("Height in Inches.")
plt.ylabel("Weight in Lbs.")
plt.show()
```

Out[27]: <matplotlib.collections.PathCollection at 0x1ce77817d00>

Out[27]: <matplotlib.collections.PathCollection at 0x1ce77817d30>

Out[27]: (51.0, 81.0, 51.0, 299.0)

Out[27]: <matplotlib.legend.Legend at 0x1ce7782b460>

Out[27]: Text(0.5, 1.0, 'Height-Weight Plotting by SUNG Mee Young')

Out[27]: Text(0.5, 0, 'Height in Inches.')

Out[27]: Text(0, 0.5, 'Weight in Lbs.')

In [28]:
```
# 평균
Gender_m=Gender.mean()
Gender_m

female_height=np.mean(fx)
female_weight=np.mean(fy)
male_height=np.mean(mx)
male_weight=np.mean(my)
print("female_height_mean : ",female_height)
print("female_weight_mean : ",female_weight)
print("male_height_mean : ",male_height)
print("male_weight_mean : ",male_weight)
```

Out[28]:

Gender	Height	Weight
Female	63.708774	135.860093
Male	69.026346	187.020621

```
female_height_mean :  63.708773603424916
female_weight_mean :  135.8600930074687
male_height_mean :  69.02634590621737
male_weight_mean :  187.0206206581929
```

In [29]:
```
# 중앙값
print("female_height_median : ",np.median(fx))
print("female_weight_median : ",np.median(fy))
print("male_height_median : ",np.median(mx))
print("male_weight_median : ",np.median(my))
```

```
female_height_median :  63.7309238591475
female_weight_median :  136.11758297008498
male_height_median :  69.02770850939555
male_weight_median :  187.033546088862
```

In [30]:
```
# 분위
print("female_height_quantile")
for i in range(0,100,25):
    print("%.2f Percentage value"%(i*0.01), quantile(fx,i*0.01))
print()
print("female_weight_quantile")
for i in range(0,100,25):
    print("%.2f Percentage value"%(i*0.01), quantile(fy,i*0.01))
print()
print("male_height_quantile")
for i in range(0,100,25):
```

```
    print("%.2f Percentage value"%(i*0.01), quantile(mx,i*0.01))
print()
print("male_weight_quantile")
for i in range(0,100,25):
    print("%.2f Percentage value"%(i*0.01), quantile(my,i*0.01))
```

```
female_height_quantile
0.00 Percentage value 54.2631333250971
0.25 Percentage value 61.89444148832923
0.50 Percentage value 63.7309238591475
0.75 Percentage value 65.5635651830946

female_weight_quantile
0.00 Percentage value 64.700126712753
0.25 Percentage value 122.93409617498699
0.50 Percentage value 136.11758297008498
0.75 Percentage value 148.8109262605865

male_height_quantile
0.00 Percentage value 58.4069049317498
0.25 Percentage value 67.17467906915428
0.50 Percentage value 69.02770850939555
0.75 Percentage value 70.98874363403955

male_weight_quantile
0.00 Percentage value 112.902939447818
0.25 Percentage value 173.887767334218
0.50 Percentage value 187.033546088862
0.75 Percentage value 200.35780180200078
```

In [31]:
```
#최빈값
print("female_height_mode :", mode(map(int,fx)))
print("female_weight_mode :", mode(map(int,fy)))
print("male_height_mode :", mode(map(int,mx)))
print("male_weight_mode :", mode(map(int,my)))
```

```
female_height_mode : [63]
female_weight_mode : [137]
male_height_mode : [69]
male_weight_mode : [192]
```

In [32]:
```
#산포도
print("female_height_data range :", data_range(fx))
print("female_weight_data range :", data_range(fy))
print("male_height_data range :", data_range(mx))
print("male_weight_data range :", data_range(my))
```

```
female_height_data range : 19.126452540972608
female_weight_data range : 137.53708702680598
male_height_data range : 20.59183741463979
male_weight_data range : 157.086759057288
```

In [33]:
```
#분산
print("female_height_variance :",np.var(fx))
print("female_weight_variance :",np.var(fy))
print("male_height_variance :",np.var(mx))
print("male_weight_variance :",np.var(my))
```

```
female_height_variance : 7.268493504171401
female_weight_variance : 361.7819105481172
male_height_variance : 8.19720348386999
male_weight_variance : 391.21581520128217
```

In [34]:
```
#표준편차
print("female_height_standard deviation :",np.sqrt(np.var(fx)))
print("female_weight_standard deviation :",np.sqrt(np.var(fy)))
print("male_height_standard deviation :",np.sqrt(np.var(mx)))
print("male_weight_standard deviation :",np.sqrt(np.var(my)))
```

```
female_height_standard deviation : 2.6960143738807107
female_weight_standard deviation : 19.020565463416624
male_height_standard deviation : 2.8630758781195427
male_weight_standard deviation : 19.779176302396472
```

In [35]:
```
#공분산
print("female_height_male_height :",np.cov(fx,mx))
print("female_weight_male_weight :",np.cov(fy,my))
print("female_height_female_weight :",np.cov(fx,fy))
print("male_height_male_weight :",np.cov(mx,my))
```

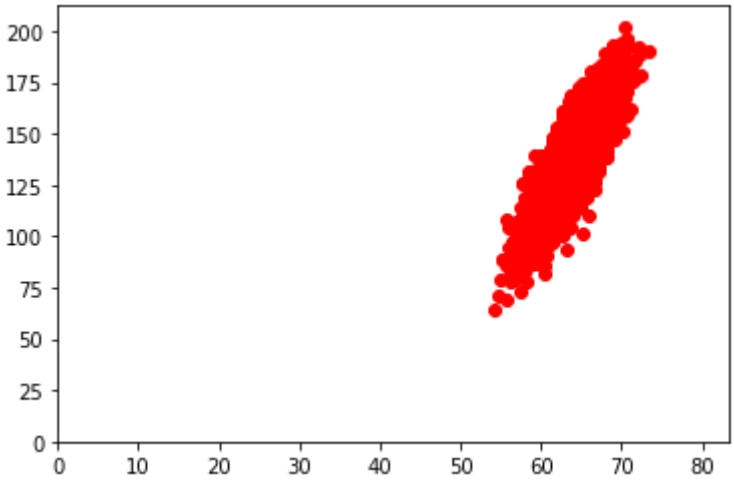
```
female_height_male_height : [[ 7.26994749 -0.25129107]
 [-0.25129107  8.19884325]]
female_weight_male_weight : [[361.8542814   -7.37151287]
 [ -7.37151287 391.29407402]]
female_height_female_weight : [[  7.26994749  43.57640416]
 [ 43.57640416 361.8542814 ]]
male_height_male_weight : [[  8.19884325  48.87964899]
 [ 48.87964899 391.29407402]]
```

In [36]:
```
#상관관계(1) - 여성의 키와 몸무게
np.corrcoef(fx,fy)

plt.plot(fx,fy,'ro')
plt.axis([0,max(fx)+10,0,max(fy)+10])
plt.show()
```

Out[36]:
```
array([[1.        , 0.84960859],
       [0.84960859, 1.        ]])
```

Out[36]:
```
[<matplotlib.lines.Line2D at 0x1ce7789cfa0>]
```

Out[36]:
```
(0.0, 83.38958586606971, 0.0, 212.237213739559)
```

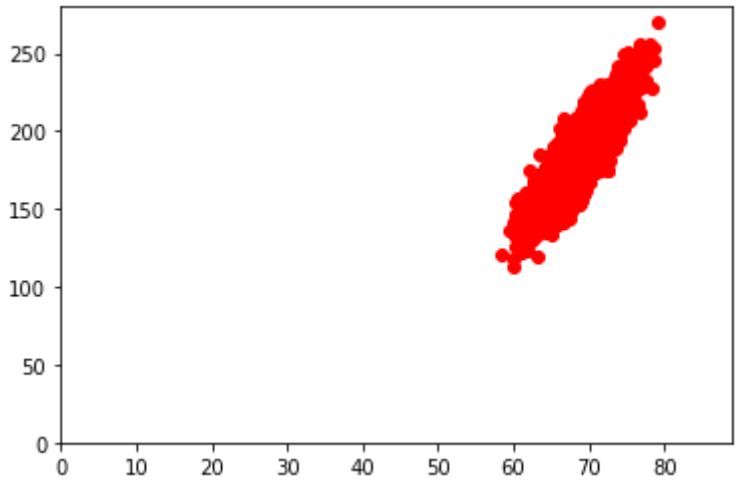


In [37]:
```
#상관관계(2) - 남성의 키와 몸무게
np.corrcoef(mx,my)

plt.plot(mx,my,'ro')
plt.axis([0,max(mx)+10,0,max(my)+10])
plt.show()
```

Out[37]:
```
array([[1.        , 0.86297885],
       [0.86297885, 1.        ]])
```

Out[37]:
```
[<matplotlib.lines.Line2D at 0x1ce778f5fd0>]
```

Out[37]:
```
(0.0, 88.99874234638959, 0.0, 279.989698505106)
```

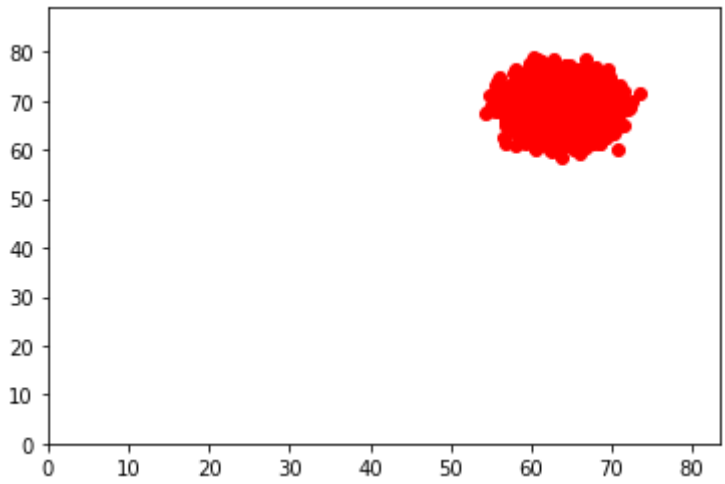


In [38]:
```
#상관관계(3) - 여성의 키와 남성의 키
np.corrcoef(fx,mx)

plt.plot(fx,mx,'ro')
plt.axis([0,max(fx)+10,0,max(mx)+10])
plt.show()
```

Out[38]:
```
array([[ 1.        , -0.03254881],
       [-0.03254881,  1.        ]])
```

Out[38]:
```
[<matplotlib.lines.Line2D at 0x1ce7892a0a0>]
```

Out[38]:
```
(0.0, 83.38958586606971, 0.0, 88.99874234638959)
```

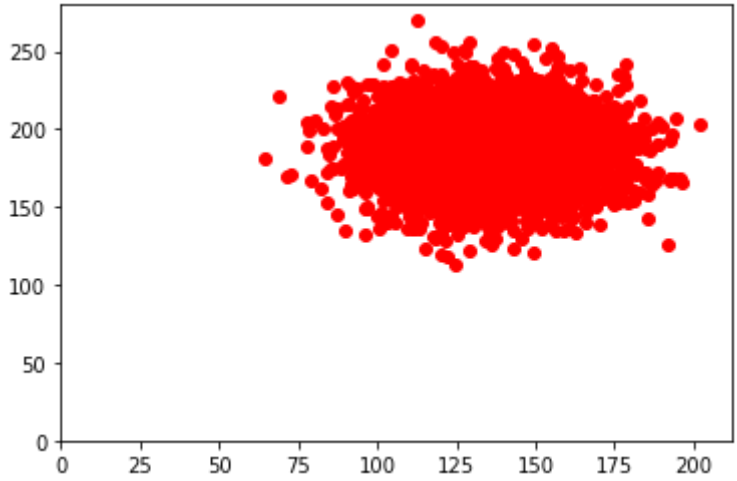


In [39]:
```
#상관관계(4) - 여성의 몸무게와 남성의 몸무게
np.corrcoef(fy,my)

plt.plot(fy,my,'ro')
plt.axis([0,max(fy)+10,0,max(my)+10])
plt.show()
```

Out[39]:
```
array([[ 1.        , -0.01959017],
       [-0.01959017,  1.        ]])
```

Out[39]:
```
[<matplotlib.lines.Line2D at 0x1ce7898c100>]
```

Out[39]:
```
(0.0, 212.237213739559, 0.0, 279.989698505106)
```



202001555 지은미