In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Exercise 1.1

1.1a) There is not evidence for causailty here. This ia observational, not experimental, data. Perhaps there is some, external, factor that both causes students to do their homework and causes them to do well on their final exam.
1.1b) Yes. You could randomly assign homework to some students but not to others. Or, more practially, randomly assign some, but not other, classes homework. The you could compare the final exam scores of students assigned homework to those not assigned homework.
1.1c) You could look at other information about students, like measures of intelligence, aptitude, conscientiousness, etc. to rule them out as causes of good final grades or to compare. Ultimately, however, you could never be sure you'd ruled out all factors besides homework that could potentially contribute to a good final exam score.

Question 2

2a

$$SSE = \sum_i^N (y_i - \mu)^2$$

$$\left.\frac{dSSE}{d\mu}\right|_{\mu^*} = \sum_i^N 2(y_i - \mu^*)(-1) = 0$$

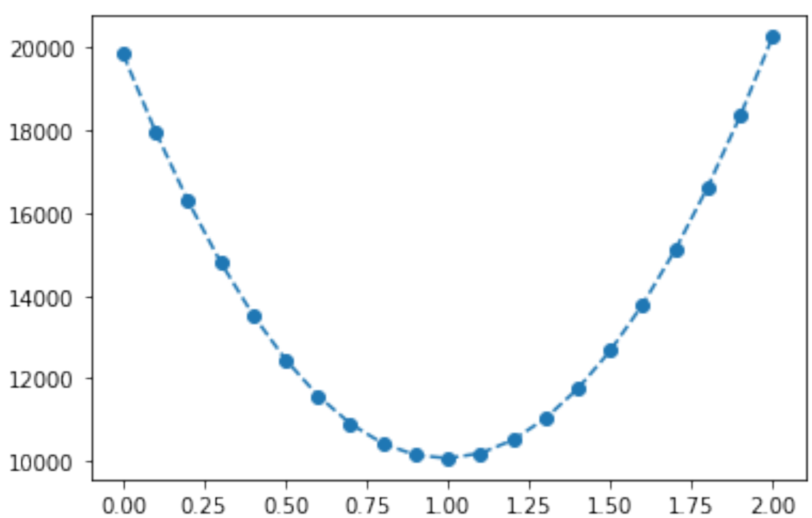
$$2\sum_i^N y_i = 2\sum_i^N \mu^* = 2N\mu^*$$

$$\mu^* = \frac{\sum_i^N y_i}{N}$$

$$\frac{d^2SSE}{d\mu^2} = 2N > 0 \Rightarrow \mu^* \text{ is a minimum}$$

2b

In [17]:
```
y = np.random.normal(1,1,size=(10000,))[:,None]
mu_candidates = np.arange(0, 2.01, .1)
sse = ((y - mu_candidates)**2).sum(axis=0)
plt.plot(mu_candidates, sse, 'o--');
```



Exercise D2.1

D2.1a

In [24]:
```
df = pd.read_csv('https://socialsciences.mcmaster.ca/jfox/Books/Applied-R
                          sep = '\s', names = ['','consumers','acres'], index

df
```

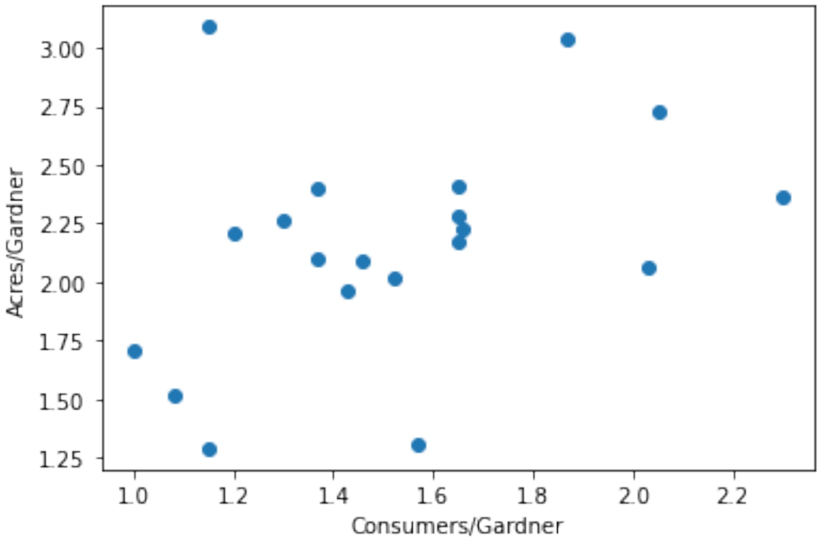
```
/Users/dsweet2/env_python3/lib/python3.6/site-packages/ipykernel_launc
her.py:2: ParserWarning: Falling back to the 'python' engine because t
he 'c' engine does not support regex separators (separators > 1 char a
nd different from '\s+' are interpreted as regex); you can avoid this
warning by specifying engine='python'.
```

Out[24]:

	consumers	acres
1	1.00	1.71
2	1.08	1.52
3	1.15	1.29
4	1.15	3.09
5	1.20	2.21
6	1.30	2.26
7	1.37	2.40
8	1.37	2.10
9	1.43	1.96
10	1.46	2.09
11	1.52	2.02
12	1.57	1.31
13	1.65	2.17
14	1.65	2.28
15	1.65	2.41
16	1.66	2.23
17	1.87	3.04
18	2.03	2.06
19	2.05	2.73
20	2.30	2.36

In [26]:
```
def scatterplot():
    x = df['consumers']
    y = df['acres']
    plt.plot(x,y,'o')
    plt.xlabel("Consumers/Gardner")
    plt.ylabel("Acres/Gardner");
```

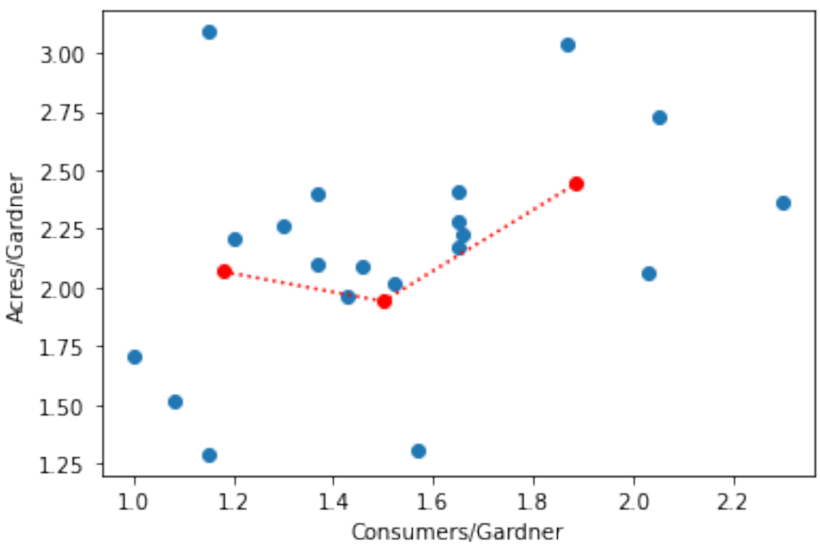
In [27]:
```
scatterplot()
```



- The relationship looks weakly and linear.
- There is a cluster of points around Consumers/Gardner ~= 1.7
- There is an outlier around the fourth point from the left (x,y) = (1.15, 3.09)

In [42]:

```
group1 = df.iloc[:7]
group2 = df.iloc[7:13]
group3 = df.iloc[13:]
means = np.array([
    group1.mean(),
    group2.mean(),
    group3.mean()
])
scatterplot()
plt.plot(means[:,0], means[:,1], 'or:');
```



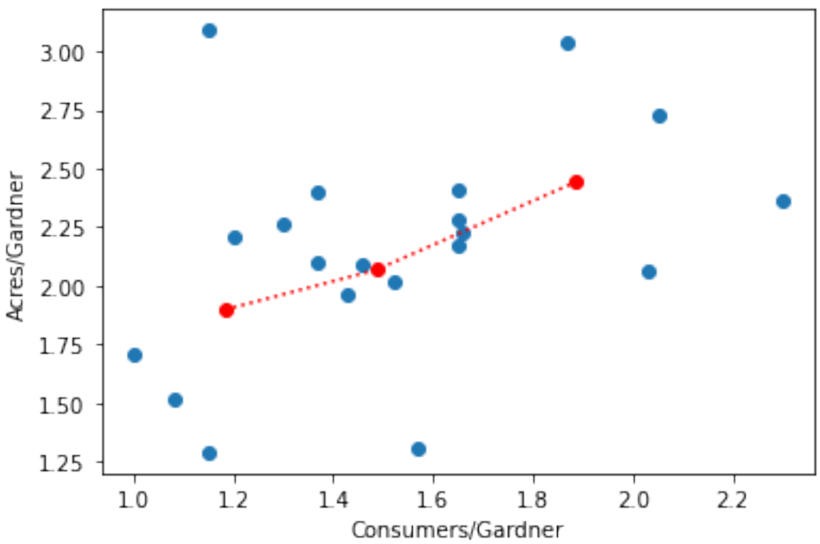
The nonparametric regression line confirms a weak, roughly linear dependence.

In [43]:

```
group1 = group1.drop(index=4)
group2 = group2.drop(index=12)
means = np.array([
    group1.mean(),
    group2.mean(),
    group3.mean()
])
scatterplot()
plt.plot(means[:,0], means[:,1], 'or:')
```

Out[43]:

```
[<matplotlib.lines.Line2D at 0x112d1a0f0>]
```



The new means make the nonparametric regression line look much more linear.

In [ ]: