Greg = 14

Marcia = 12

Peter = 11

Jan = 10

Bobby = 8

Cindy = 6

Cousin Oliver = 8

Calculations by hand:

Mean = (14 + 12 + 11 + 10 + 8 + 6 + 8) / 7 = 9.857

Median = 10

Mode = 8

Variance = ((14-9.857)^2 + (12-9.857)^2 + (11-9.857)^2 + (10-9.857)^2 + (8-9.857)^2 + (6-9.857)^2 + (8-9.857)^2) / (7) = 6.408

Standard Deviation = 7.476^0.5 = 2.531

Standard Error = 2.734/(7^0.5) = 1.033

In [1]:
```
import numpy as np
import pandas as pd
```

In [2]:
```
df = pd.DataFrame()

df['Name'] = ['Greg', 'Marcia', 'Peter', 'Jan', 'Bobby', 'Cindy', 'Cousin Oliver']
df['Age'] = [14, 12, 11, 10, 8, 6, 8]
```

In [3]:
```
#Calculate mean, median, mode, variance, standard deviation, standard error
mean = np.mean(df.Age)
print(mean)

median = np.median(df.Age)
print(median)

(values, counts) = np.unique(df['Age'], return_counts=True)
ind = np.argmax(counts)
mode = values[ind]
print(mode)

variance = np.var(df['Age'])
print(variance)

standard_deviation = np.std(df['Age'])
print(standard_deviation)

standard_error = np.std(df['Age']) / np.sqrt(len(df['Age']) -1)
print(standard_error)
```

```
9.857142857142858
10.0
8
6.408163265306122
2.531435020952764
1.0334540197243194
```

For a measure of central tendency, I would use the median value of 10. Since we are examining the age for a small group of individuals, all of which are relatively close to the same age (no outliers), using the median which is a nice whole number seems to make the most sense. For a measure of variance, I would use the standard deviation. This shows the variation in the dataset and how close data points are to the expected value.

In [4]:
```
#Cindy has a birthday
df.loc[[5], 'Age'] = 7

#Recaculate mean, median, mode, variance, standard deviation, standard error
mean = np.mean(df.Age)
print(mean)

median = np.median(df.Age)
print(median)

(values, counts) = np.unique(df['Age'], return_counts=True)
ind = np.argmax(counts)
mode = values[ind]
print(mode)

variance = np.var(df['Age'])
print(variance)

standard_deviation = np.std(df['Age'])
print(standard_deviation)

standard_error = np.std(df['Age']) / np.sqrt(len(df['Age']) -1)
print(standard_error)
```

```
10.0
10.0
8
5.428571428571429
2.32992949004287
0.951189731211342
```

After Cindy's birthday, not much changed with the measures of central tendency. The median and mode stayed the same as expected and the mean increased only slightly from 9.857 to 10. The measures of variance all decreased as expected. Since Cindy was the youngest, her increase in age reduced the variability of the data. If Greg would have had a birthday instead of Cindy on the otherhand, variability would have increased for the data since he is the oldest.

In [5]:
```
#Replace Cousin Oliver with Jessica
df.loc[[6], 'Name'] = 'Jessica'
df.loc[[6], 'Age'] = 1

mean = np.mean(df.Age)
print(mean)

median = np.median(df.Age)
print(median)

(values, counts) = np.unique(df['Age'], return_counts=True)
ind = np.argmax(counts)
mode = values[ind]
print(mode)

variance = np.var(df['Age'])
print(variance)

standard_deviation = np.std(df['Age'])
print(standard_deviation)

standard_error = np.std(df['Age']) / np.sqrt(len(df['Age']) -1)
print(standard_error)
```

```
9.0
10.0
1
15.428571428571429
3.927922024247863
1.6035674514745466
```

Adding Jessica, the 1 year old, and removing Cousin Oliver does not change my choice of standard deviation as a measure of variation. I would also keep the median as my choice to measure central tendency. With Jessica being a bit of an outlier, I think using the median is now more representative of the group's age. The mean is now slightly more skewed.

Looking at the polling percentages I would average the three mainstream magazines (TV Guide, Entertainment Weekly and Pop Culture Today) to estimate that about 20% of America was fans of the Brady Bunch on its 50th anniversary. The other magazine, SciPhi Phanatic is probably geared toward a targeted audience, one that is not representative of the entire population. For this reason, I think it is fair to say there is bias in the polling results from that sample and we should disregard them.