Imagine we do an experiment with two groups (an experimental and a control). In this experiment, we have 20 people in each condition.

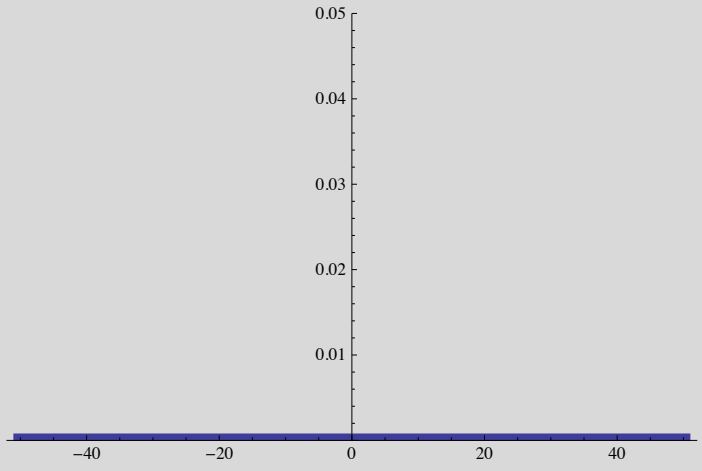
```
Needs["Histograms`"];
Needs["HypothesisTesting`"];
SetDirectoryToNotebookLocation[];
```

```
1 + 1
```

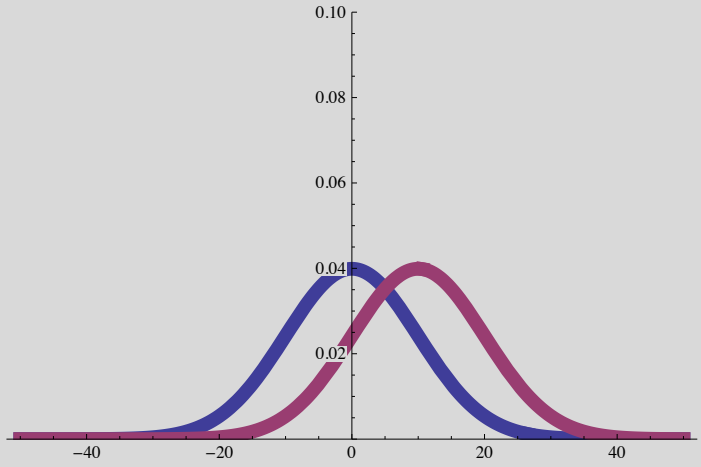
```
2
```

```
ndist = NormalDistribution[0, 10];
ndist2 = NormalDistribution[10, 10];
```

```
f = PDF[ndist, x];
g = PDF[ndist2, x];
myplot2 =
 Plot[{0}, {x, -50, 50}, {PlotRange → {0, 0.05}, PlotStyle → {{Thickness[0.02`]}}}]
```



```
f = PDF[ndist, x];
g = PDF[ndist2, x];
myplot2 =
 Plot[{f, g}, {x, -50, 50}, {PlotRange → {0, 0.1}, PlotStyle → {{Thickness[0.02`]}}}]
```




```
groupa = Table[Random[ndist], {20}];
groupb = Table[Random[ndist2], {20}];
```

Let's look at our data:

```
mydata = Transpose[{groupa, groupb}];
mydata // TableForm
```

```
-11.6082  2.58565
-12.7528  6.0877
12.0938   6.30211
-2.2291   13.1693
2.68462   29.4261
9.23352   6.29066
9.92366   -1.25028
-13.1257  -8.80264
-2.86888  30.3614
0.868362  -11.9505
14.3565   -0.633046
13.6586   21.7459
-6.41646  26.6926
-6.92003  2.53062
-5.70274  4.90664
6.73401   -7.30321
-13.1511  14.2896
20.7136   14.0744
-5.65129  10.8784
-8.64905  6.12983
```

Are these groups different? How can we tell?

One way is to begin to describe the data. How about if we consider the maximum value and minimum value in each group?

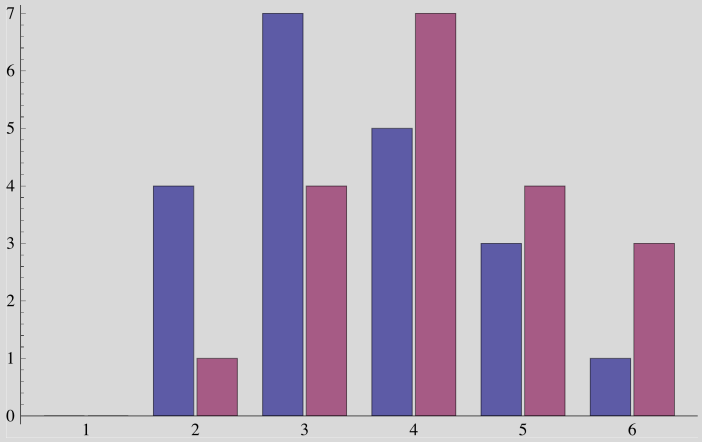
```
mydata // TableForm
```

```
-11.6082  2.58565
-12.7528  6.0877
12.0938   6.30211
-2.2291   13.1693
2.68462   29.4261
9.23352   6.29066
9.92366   -1.25028
-13.1257  -8.80264
-2.86888  30.3614
0.868362  -11.9505
14.3565   -0.633046
13.6586   21.7459
-6.41646  26.6926
-6.92003  2.53062
-5.70274  4.90664
6.73401   -7.30321
-13.1511  14.2896
20.7136   14.0744
-5.65129  10.8784
-8.64905  6.12983
```

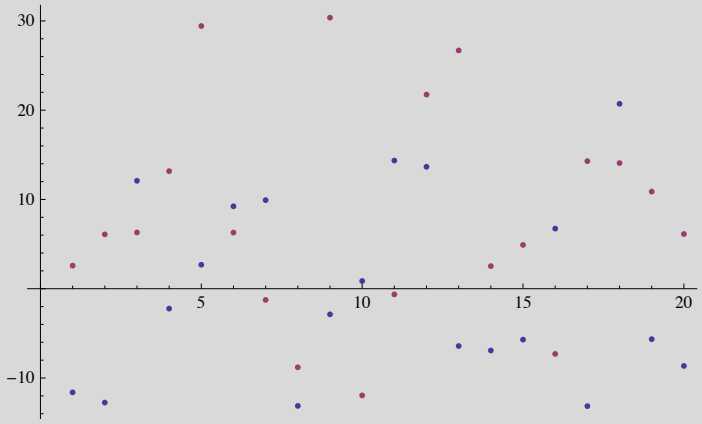
```
Transpose[{Sort[groupa], Sort[groupb]}] // TableForm
```

```
-13.1511  -11.9505
-13.1257  -8.80264
-12.7528  -7.30321
-11.6082  -1.25028
-8.64905  -0.633046
-6.92003  2.53062
-6.41646  2.58565
-5.70274  4.90664
-5.65129  6.0877
-2.86888  6.12983
-2.2291   6.29066
0.868362  6.30211
2.68462   10.8784
6.73401   13.1693
9.23352   14.0744
9.92366   14.2896
12.0938   21.7459
13.6586   26.6926
14.3565   29.4261
20.7136   30.3614
```

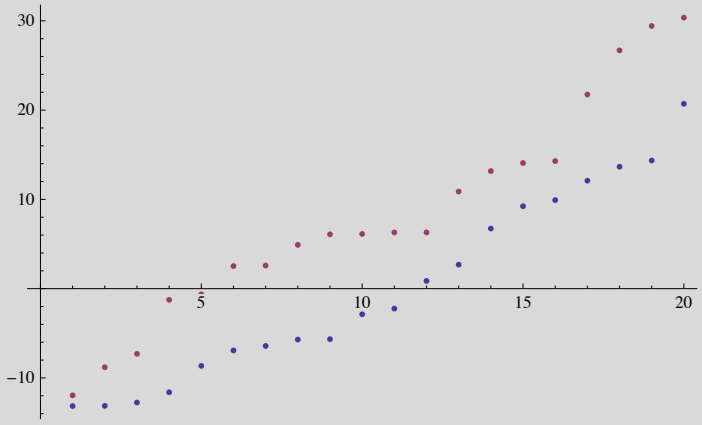
```
BarChart[BinCounts[groupa, {-30, 30, 10}], BinCounts[groupb, {-30, 30, 10}]]
```




```
ListPlot[{groupa, groupb}]
```




```
ListPlot[{Sort[groupa], Sort[groupb]}]
```



```
{Max[groupa], Min[groupa]}
{Max[groupb], Min[groupb]}
```

```
{20.7136, -13.1511}
```

```
{30.3614, -11.9505}
```

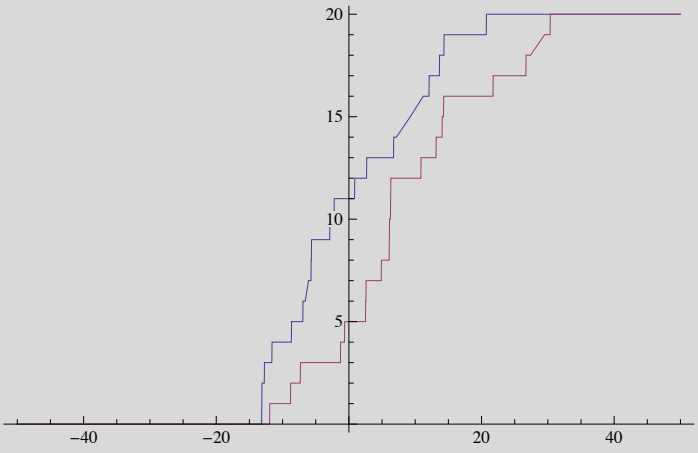
How about the RANGE of the data (max-min)?

```
Max[groupa] - Min[groupa]
Max[groupb] - Min[groupb]
```

```
33.8646
```

```
42.312
```

```
Plot[
 {Length[Select[Sort[groupa], # < x &]],
  Length[Select[Sort[groupb], # < x &]]}, {x, -50, 50}]
```



Measures of CENTRAL TENDENCY

We begin our discussion with the desire to describe a collection of numbers. It just so happens that there is two main ways in which to describe a collection of numbers. One is called the measure of central tendency and another is the measure of dispersion or scatter. We will first look at a number of different measure of each of these.

Mean

In more concrete terms, the mean is quite simply defined as the the sum of all the scores in a sample divided by the number of scores in the sample.

$$\overline{X} = \frac{\sum_i x_i}{n}$$

There are a number of key points about the mean, First is that the mean is the score around which the deviation scores sum to zero. We can take a look at this informally by just making up a random sample, finding the mean of this sample, and finding the sum of the deviation scores around that mean.

Here we generate 10 random numbers between 0 and 100.

```
samples = RandomReal[{0, 100}, 10]
```

```
{60.117, 73.7687, 14.6072, 47.3007, 73.9176, 93.7031, 56.0971, 68.5044, 50.3203, 86.0446}
```

We can find the mean quite easily using the equation (1). We can check this with *Mathematica's* built in Mean[ ] function. First, we can find the sum of all the samples.

```
sumsamples = Apply[Plus, samples]
meanbyhandsample = sumsamples / Length[samples]
```

```
624.381
```

```
62.4381
```

Now, we can check this with *Mathematica*.

```
meansamples = Mean[samples]
```

```
62.4381
```

```
meansamples == meanbyhandsample
```

```
True
```

Now that we know the mean, lets find the deviations from the mean for each sample. What we to know is how far each indivdual score is from the mean. Fortunately, this is very simple to express in *Mathematica*.

```
deviations = samples - meansamples
```

```
{-2.32103, 11.3306, -47.8309, -15.1374,
 11.4795, 31.265, -6.34099, 6.06631, -12.1177, 23.6065}
```

Now lets add up all the deviations to see if the do indeed sum to zero.

```
Apply[Plus, deviations]
```

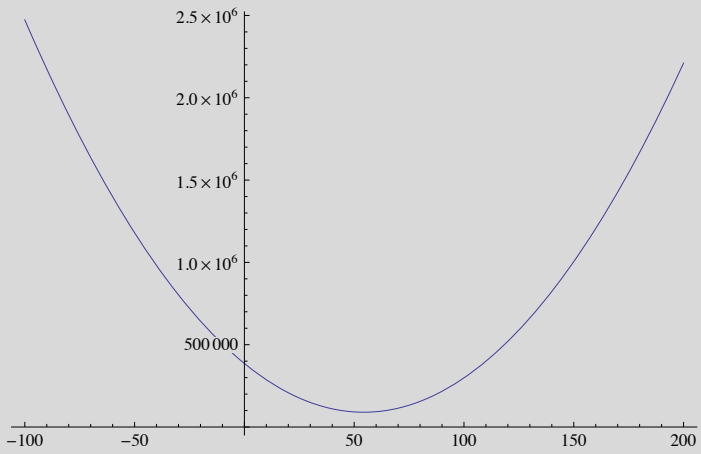
$$-2.84217 \times 10^{-14}$$

Well, it is not quite zero, but that is because *Mathematica* has quite a bit of precision, but anything to the power of -14 is close enough to zero for us.

A second important property of the mean is that it is the score for which the squared deviations of the sample is a minimum. Imagine we were to find the deviations for each sample from the mean, and square it, then add these numbers up. What this property of the mean "means" is that the mean will minimize the squared deviations value.

We can take a look at this also. Lets take another random sample of 100 scores this time and find the sum of squared deviations for a range of values. We will see that if we choose the mean this value will be at a minimum.

```
Clear[samples]
Clear[f]
Clear[func]
samples = RandomReal[{0, 100}, 100];
f[x_] := Plus @@ (samples - x)^2;
func = Plot[f[x], {x, -100, 200}]
```



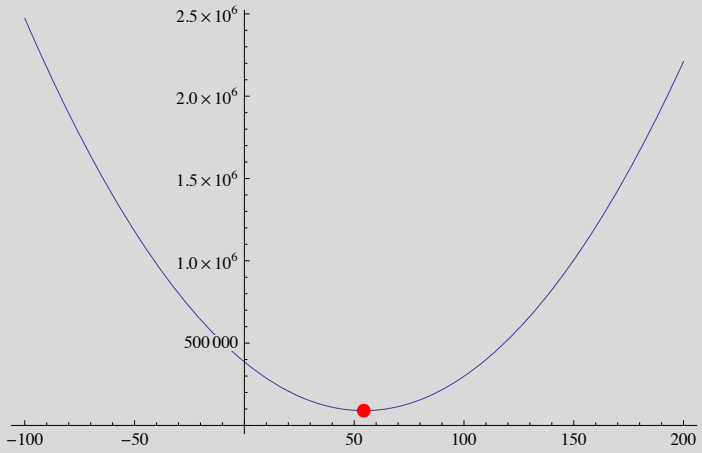
So we can see that the sum of squared deviation looks like a regular parabola which a minimum value somewhere around 50. Well, we can figure out exactly what the minimum of this function is by finding the mean of our sample.

```
Clear[samplemean]
samplemean = Mean[samples]
```

```
54.3759
```

If we plot this point on the graph we can see that it is in fact the minimum.

```
Clear[point]
point = ListPlot[{{samplemean, f[samplemean]}},
    PlotStyle → {PointSize[0.02`], Red}, DisplayFunction → Identity];
Show[{func, point}, DisplayFunction → $DisplayFunction]
```



The big red dot is the mean superimposed on the function of the sum of squared deviaions.

A final note about the mean is that it is sensitive to outliers in the data. We should all be aware of how one exceptional student can blow the curve for the whole class. It is important in statistical analysis to take onliers out under only extremely justified reasons. Lets take a quick look at an example of this. We are going to repeat this experiment with the median in the mode in the next section so we can compare the relavtive sensitivity of the mean, median, and mode to outliers.

```
Clear[samples];
samples = RandomReal[{0, 100}, 100];
Mean[samples]
```

```
51.3902
```

First we find the mean of 100 random numbers. Then we can add one really crazy outlier, 1000 and see how it affects the mean.

```
AppendTo[samples, 1000];
Mean[samples]
```

```
60.7823
```

Wow, if shot up quite a bit! In the next sections we can see how sensitive the median and mode are to this type of phenomena in our data.

Median

The median is the "middle" score. Half the scores in your sample will be above the median and half will be below it.

Lets take a look a 10 random number and find the median score. Five of the numbers should be above the median and 5 should be below it.

```
samples = RandomReal[{0, 100}, 11];
mymedian = Median[samples]
sorted = Sort[samples]
```

```
37.5484
```

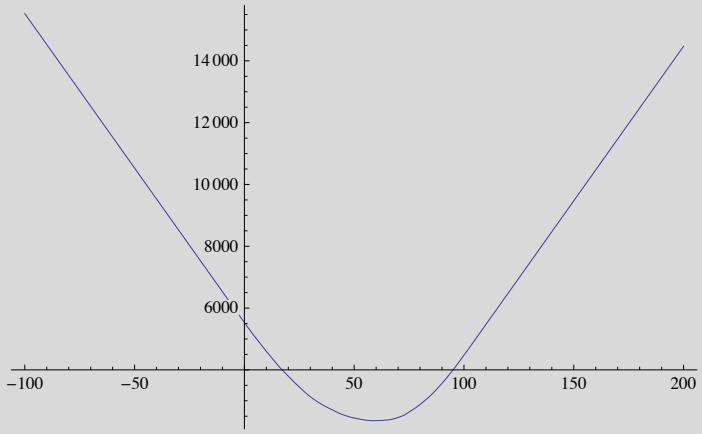
```
{20.9391, 26.6797, 27.3063, 29.3753, 34.3301,
 37.5484, 66.2146, 77.6886, 78.8282, 85.7355, 98.2178}
```

As we can see from the sorted version of the list this is true. The only thing to remember about the median is that if there is a even number of scores in your sample, then the median is defined as the average of the two middle scores. In this case, the median is the score half way between the 5th and 6th sorted score. We can check to verify.

```
median = (sorted[[5]] + sorted[[6]]) / 2
median == mymedian
```

The median minimizes the sum of the absolute deviations around itself. In the same way as we showed that the mean minimizes the sum of squared deviations of the sample, we can show that the median does the same for the absolute deviations. (the absolution deviations are the absolute value of the deviation around a score.)

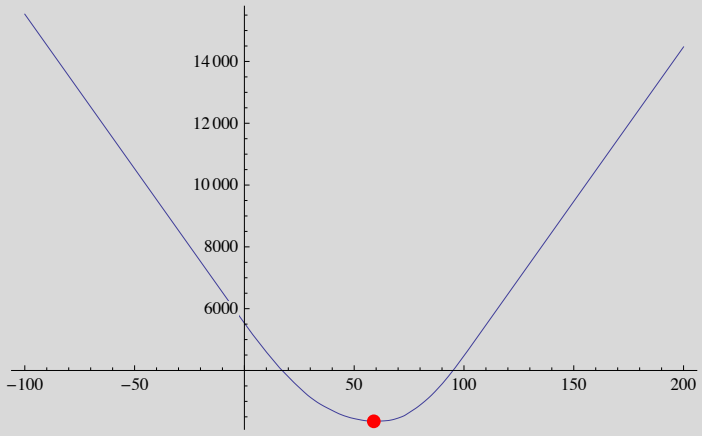
```
Clear[samples]
Clear[f]
Clear[func]
samples = RandomReal[{0, 100}, 100];
f[x_] := Plus @@ Abs[samples - x];
func = Plot[f[x], {x, -100, 200}]
```




```
samplemedian = Median[samples]
```

```
58.9318
```

```
Clear[point]
point = ListPlot[{{samplemedian, f[samplemedian]}},
    PlotStyle → {PointSize[0.02`], Red}, DisplayFunction → Identity];
Show[{func, point}, DisplayFunction → $DisplayFunction]
```



As we can see, once again, the median is the minimum value of the function devided by the sum of squared deviations.

So how sensitive is the median to outliers. Lets perform our ourlier test and see.

```
Clear[samples];
samples = RandomReal[{0, 100}, 100];
Median[samples]
```

```
49.7976
```

```
AppendTo[samples, 1000];
Median[samples]
```

```
49.8329
```

So the median went up by just about 1 whereas the mean went up by 20 to a outlier at 1000!! The median is thus less affected by extreme scores than the mean.

Mode

The mode is simple the most frequent score. If you wanted to guess an individual score and wanted to be exactly right the most, guess the mode. Otherwise the mode isn't that interesting.

Measures of DISPERSION OR SCATTER

Generally, we consider the measure of central tendency to be the overall magnitude of socres in the group of scores. This might be something like the DC level in a sinusoid. Individual scores in our sample will be distributed around this measure of central tendency. Thus, we can use the average distance of scores away from the mean as our measure of dispersion. If we use the mean as our measure of central tendency then this average will obviously be zero. Instead we can square each deviation score and then take the average. This value is called the variance or mean squared deviation score.

Variance

$$\sigma^2 = \frac{\sum_i (x_i - \bar{x})^2}{n - 1}$$

$$s^2 = \frac{\sum_i (x_i - \bar{x})^2}{n - 1}$$

```
randomlist = RandomReal[{0, 10}, 10]
```

```
{7.31347, 4.88155, 2.24012, 8.07227,
 1.61297, 7.38737, 2.18041, 7.87216, 1.37213, 4.78341}
```

```
Apply[Plus, (randomlist - Mean[randomlist])^2]
            Length[randomlist] - 1
Variance[randomlist]
```

```
7.60883
```

```
7.60883
```

Standard Deviation

The variance is great but the units of it are the squared values of our observations in the sample. For example, if we had a collection of lengths (in meters, m), then the variance would be expressed as m^2. Instead, we can take the square root of the variance to get what is called the standard deviation.

$$\sigma = \sqrt{\sigma^2} = \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{n - 1}}$$

$$s = \sqrt{s^2} = \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{n - 1}}$$

```
Sqrt[Apply[Plus, (randomlist - Mean[randomlist])^2]/(Length[randomlist] - 1)]
StandardDeviation[randomlist]
```

```
2.75841
```

```
2.75841
```

Another measure is the average deviation. To calculate this value, we take the absolute value of each deviation score. This is the true average "distance" of each point in our sample from the mean.

```
Apply[Plus, Abs[randomlist - Mean[randomlist]]]/Length[randomlist]
```

```
2.30942
```

Mathematica provides a nice function called the DispersionReport[] which calculates a number of dispersion measures at once.

```
DispersionReport[randomlist]
```

```
{Variance → 10.2239, StandardDeviation → 3.19749, SampleRange → 9.14665,
 MeanDeviation → 2.30942, MedianDeviation → 1.28867, QuartileDeviation → 1.28867}
```

A final measure of dispersion is the standard error of the mean which is calculated by dividing the standard deviation by the square root of the number of values in the sample.

```
StandardDeviation[randomlist] / Sqrt[Length[randomlist]]
StandardErrorOfSampleMean[randomlist]
```

```
1.01113
```

```
1.01113
```

Combining information from two samples

If we want to combine information from two groups drawn from the same population, we can use the following formula for what is known as the "pooled" mean and "pooled" variance. Basically, they just reflect the combined mean of the two group. However, instead of going back and adding everything up again, you can just use the size of each group (n) and the mean/std. dev. to compute it.

Pooled Means

$$\overline{x_{12}} = \frac{n_1 \overline{x_1} + n_2 \overline{x_2}}{n_1 + n_2}$$

For example, lets make two random samples and verify the above:

```
sample1 = RandomReal[{0, 1}, 10];
sample2 = RandomReal[{0, 1}, 10];
bothsamples = Flatten[Append[sample1, sample2]];
```

```
(Mean[sample1] - Mean[sample2]) /
  (Sqrt[(Variance[sample1] + Variance[sample2]) / 2] * Sqrt[2 / 10])
```

```
0.197409
```

```
MeanTest[sample1 - sample2, 0, FullReport → True]
MeanDifferenceTest[sample1, sample2, 0, FullReport → True, EqualVariances → True]
```

```
{FullReport → Mean      TestStat  Distribution
              0.030678  0.216568  StudentTDistribution[9]
, OneSidedPValue → 0.416687}
```

```
{FullReport → MeanDiff  TestStat  Distribution
              0.030678  0.197409  StudentTDistribution[18]
, OneSidedPValue → 0.42286}
```

```
Mean[sample1]
```

```
0.471538
```

```
Mean[sample2]
```

```
0.44086
```

```
Mean[bothsamples]
```

```
0.456199
```

```
Mean[sample1] + Mean[sample2]
                2
```

```
0.456199
```

You see that when the size of the same (n's) the pooled mean is just the average of the two because:

$$\overline{x_{12}} = \frac{n\,\overline{x_1} + n\,\overline{x_2}}{n+n} = \frac{n(\overline{x_1}+\overline{x_2})}{2\,n} = \frac{(\overline{x_1}+\overline{x_2})}{2}$$

You see that when the size of the same (n's) the pooled mean is just the average of the two because:

$$t = \frac{\overline{x_1} - \overline{x_2}}{s/\sqrt{n}}$$

$$t = \frac{\overline{x_1} - \overline{x_2}}{s^2/n}$$

$$t = \frac{\overline{x_1} - \overline{x_2}}{s^2\left(\frac{1}{n}\right)}$$

$$t = \frac{\overline{x_1} - \overline{x_2}}{s_{12}{}^2\left(\frac{1}{n_1} + \frac{1}{n_2}\right)}$$

$$t = \frac{\overline{x_1} - \overline{x_2}}{s_{12}\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$t = \frac{\sqrt{\frac{n}{2}}\ (\overline{x_1} - \overline{x_2})}{s_{12}}$$

$$t = \frac{\frac{n}{2}\ (\overline{x_1} - \overline{x_2})^2}{s_{12}{}^2}$$

We might be tempted to do the same experiment with the variance/standard deviation:

$$s_{12}{}^2 = \frac{(n_1 - 1)\ s_1{}^2 + (n_2 - 1)\ s_2{}^2}{(n_1 - 1) + (n_2 - 1)}$$

if n is equal we get

$$s_{12}^2 = \frac{(n-1)\,s_1^2+(n-1)\,s_2^2}{(n-1)+(n-1)}$$

$$= \frac{(n-1)\left(s_1^2+s_2^2\right)}{2\,(n-1)}$$

$$= \frac{\left(s_1^2+s_2^2\right)}{2}$$

but let's try:

```
(Variance[sample1] + Variance[sample2])/2 == Variance[bothsamples]
```

```
False
```

Since the variances in the case described above are the moments calculated around their own respective sample means, it doesn't match the variance for the two samples treated as one.

```
(Variance[sample1] + Variance[sample2])/2
Variance[bothsamples]
```

```
0.120751
```

```
0.114643
```