This is the basis for my $100 bet used in one of my February 2012 blog posting. The below is not well documented. However, if you follow the blog entry it may make more sense. Here are a couple of references that I found helpful:

http://science.kennesaw.edu/~jdemaio/1107/Central%20Limit%20Theorem.htm
http://www.statisticalengineering.com/central_limit_theorem_(summary).html

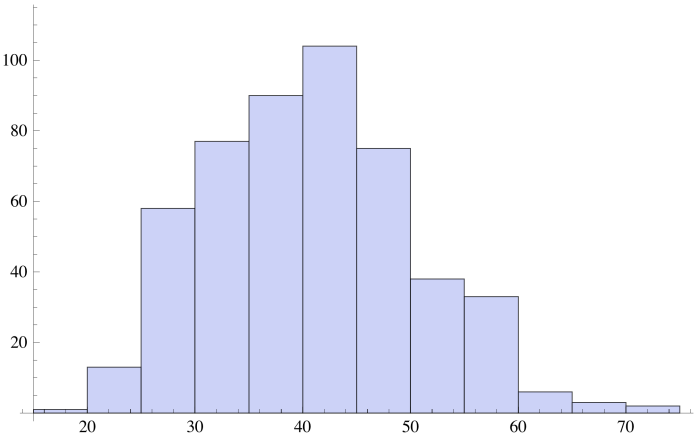
Enjoy! Craig on 23-Feb-2012.

First, create the master population. We won't use all these values and the actual population of values is shown at the very bottom of the file. Notice I used a Chi Square distribution. I wanted to use something that was not "normal" yet seemd real enough to mimic real DBA ages. The normality test showed a pvalue < 0.05, meaning the samples are NOT likely to be normally distributed. The histogram looks pretty normal though.

In[1]:=
```
SeedRandom[41]; (* used 41 for blog posting w/avg age of 40 *)
sampleSetSize = 500;
allSamples = RandomVariate[ChiSquareDistribution[40], sampleSetSize];
DistributionFitTest[allSamples]
Histogram[allSamples]
```

Out[3]= 0.0142331

Out[4]=



Walk into a room full of Oracle DBAs (perhaps 300). But this is kind of lame... I should have skipped this sub setting the raw data. Must have seemed good at the time.

In[5]:=
```
start = 3; (* 3 for blog posting *)
fullSet = Take[allSamples, {start, start + 300}]
N[Min[fullSet]]
N[Mean[fullSet]]
N[Median[fullSet]]
N[Max[fullSet]]
N[StandardDeviation[fullSet]]
```

Out[5]=
```
{44.5757, 30.785, 35.6868, 29.9242, 50.1605, 26.0015, 30.1081, 34.0855, 43.95, 39.1439,
 44.839, 33.9597, 41.7853, 46.8476, 38.4352, 36.8413, 57.4797, 45.0145, 48.7167, 31.2964,
 38.2674, 26.8536, 58.0649, 51.6393, 35.5164, 32.279, 47.7199, 42.5412, 36.0783,
 65.8272, 48.8191, 45.4366, 46.7295, 52.8966, 32.1351, 37.9589, 27.1767, 58.2877,
 30.8401, 32.0744, 50.4045, 33.4604, 47.9038, 32.8775, 39.3134, 52.503, 38.8376,
 50.3528, 48.8238, 58.6698, 42.7621, 35.2887, 54.3402, 35.1732, 38.0354, 42.4413,
 33.484, 40.8252, 43.3936, 37.4256, 39.854, 56.5796, 38.3062, 51.7921, 41.6249, 48.1903,
 59.5524, 25.6498, 30.212, 27.7241, 44.0799, 40.1091, 32.3379, 38.4909, 51.2488, 42.2192,
 41.0797, 42.6949, 31.957, 36.5064, 57.8382, 48.4274, 51.8952, 43.4507, 29.4731,
 70.9667, 51.5625, 33.8341, 28.2977, 45.2365, 40.9395, 37.3965, 29.1383, 60.1161,
 48.5472, 40.7297, 28.7761, 43.0367, 47.5859, 35.5642, 38.707, 33.6054, 42.1052,
 56.8194, 40.7146, 48.6739, 27.2828, 46.7747, 29.5329, 27.1862, 62.3574, 30.5472,
 58.3563, 47.6477, 31.6724, 54.662, 47.0582, 40.88, 47.826, 35.9465, 41.6942, 40.2792,
 44.2132, 29.8962, 43.6436, 44.8206, 36.7735, 57.636, 29.6445, 46.0112, 37.1612,
 34.0818, 36.6415, 45.7838, 43.0134, 36.0074, 25.208, 28.0371, 49.452, 38.9837, 38.7388,
 48.5738, 41.1468, 39.711, 42.4779, 48.3371, 47.5775, 55.857, 41.4958, 55.0787, 40.204,
 48.6906, 56.636, 39.0096, 34.4788, 33.7795, 27.5248, 28.2069, 44.7196, 31.9433,
 55.7008, 24.6043, 53.7665, 35.3491, 48.1037, 45.845, 32.6105, 48.0552, 40.0636,
 35.2186, 29.457, 55.4423, 39.2176, 28.2254, 44.3722, 71.3469, 38.6211, 27.8446,
 42.2412, 37.5354, 38.7751, 54.6499, 38.4456, 35.2427, 28.9652, 43.3554, 50.7664,
 28.9852, 42.0611, 43.5633, 52.3444, 31.9971, 28.4177, 37.1, 43.9791, 62.6144, 34.5215,
 57.0141, 23.6159, 41.1221, 39.3029, 51.2828, 41.6449, 32.2892, 45.9017, 53.3298,
 35.2187, 35.9236, 42.0673, 44.5298, 51.8486, 60.584, 28.9184, 40.53, 51.0339, 33.6537,
 40.7803, 58.461, 40.2694, 30.0798, 47.0634, 40.9521, 33.5133, 36.6249, 44.1324, 33.568,
 34.9512, 34.822, 45.7616, 34.5336, 40.2951, 43.9616, 42.5516, 54.7211, 40.9132, 36.5334,
 41.7575, 39.4575, 41.1793, 25.6888, 37.1895, 24.554, 55.7841, 34.9596, 42.9726,
 37.7639, 36.2597, 34.5336, 31.4729, 26.8748, 38.0113, 23.0372, 43.4735, 29.9629,
 45.7797, 40.5757, 45.2938, 43.8257, 33.2357, 39.6969, 29.7761, 29.0498, 29.3406,
 57.1073, 40.0078, 47.2877, 38.6576, 22.566, 46.8678, 28.779, 28.8258, 30.5364, 45.5216,
 39.5363, 60.8583, 56.2008, 25.4158, 45.6375, 30.1801, 31.5807, 36.2699, 40.3521,
 26.1184, 43.0147, 46.3911, 39.0053, 26.3804, 52.0422, 47.8746, 43.6934, 42.4077,
 41.619, 37.2735, 32.2332, 32.017, 32.8249, 47.39, 46.8228, 43.747, 49.6794, 35.242}
```

Out[6]= 22.566

Out[7]= 40.933

Out[8]= 40.53

Out[9]= 71.3469

Out[10]= 9.36128

Ask 25 of the DBAs to walk to the other side of the room. Of those 25, randomly divide them into 5 groups of 5. Ask the others to remain in the room. This is the real data for the initial 5 sample sets.

In[11]:=
```
initialSample = fullSet[[1 ;; 100]]
set1 = initialSample[[1 ;; 5]]
set2 = initialSample[[6 ;; 10]]
set3 = initialSample[[11 ;; 15]]
set4 = initialSample[[16 ;; 20]]
set5 = initialSample[[21 ;; 25]]
```

Out[11]=
```
{44.5757, 30.785, 35.6868, 29.9242, 50.1605, 26.0015, 30.1081, 34.0855, 43.95, 39.1439,
 44.839, 33.9597, 41.7853, 46.8476, 38.4352, 36.8413, 57.4797, 45.0145, 48.7167,
 31.2964, 38.2674, 26.8536, 58.0649, 51.6393, 35.5164, 32.279, 47.7199, 42.5412,
 36.0783, 65.8272, 48.8191, 45.4366, 46.7295, 52.8966, 32.1351, 37.9589, 27.1767,
 58.2877, 30.8401, 32.0744, 50.4045, 33.4604, 47.9038, 32.8775, 39.3134, 52.503,
 38.8376, 50.3528, 48.8238, 58.6698, 42.7621, 35.2887, 54.3402, 35.1732, 38.0354,
 42.4413, 33.484, 40.8252, 43.3936, 37.4256, 39.854, 56.5796, 38.3062, 51.7921,
 41.6249, 48.1903, 59.5524, 25.6498, 30.212, 27.7241, 44.0799, 40.1091, 32.3379,
 38.4909, 51.2488, 42.2192, 41.0797, 42.6949, 31.957, 36.5064, 57.8382, 48.4274,
 51.8952, 43.4507, 29.4731, 70.9667, 51.5625, 33.8341, 28.2977, 45.2365, 40.9395,
 37.3965, 29.1383, 60.1161, 48.5472, 40.7297, 28.7761, 43.0367, 47.5859, 35.5642}
```

Out[12]=
```
{44.5757, 30.785, 35.6868, 29.9242, 50.1605}
```

Out[13]=
```
{26.0015, 30.1081, 34.0855, 43.95, 39.1439}
```

Out[14]=
```
{44.839, 33.9597, 41.7853, 46.8476, 38.4352}
```

Out[15]=
```
{36.8413, 57.4797, 45.0145, 48.7167, 31.2964}
```

Out[16]=
```
{38.2674, 26.8536, 58.0649, 51.6393, 35.5164}
```

Ask each small group of 5 to determine their average age, with the precision to 1 decimal place, tell me the values. This is where I got the initial five values I wrote about in the blog.

In[17]:=
```
mean1 = N[Mean[set1]]
mean2 = N[Mean[set2]]
mean3 = N[Mean[set3]]
mean4 = N[Mean[set4]]
mean5 = N[Mean[set5]]
```

Out[17]=
```
38.2265
```

Out[18]=
```
34.6578
```

Out[19]=
```
41.1734
```

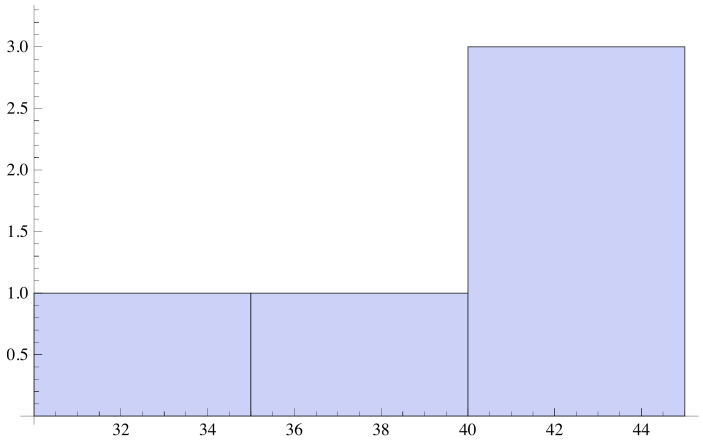
Out[20]=
```
43.8697
```

Out[21]=
```
42.0683
```

Below is a histogram of the mean sample set. It's doesn't look normal, but that will be difficult with only five values. The normality test however clearly exceeded our pvalue threshold of 0.05.

In[22]:=
```
meanSet = {mean1, mean2, mean3, mean4, mean5};
Histogram[meanSet]
```

Out[23]=



In[24]:=
```
DistributionFitTest[meanSet]
```

Out[24]=
```
0.859285
```

Since the sample set of the means is normal, I can make all sorts of claims. But I need to calculate it's mean (i.e., the mean of the sample set of means), standard deviation, and the 90% and 95% confidence intervals.

In[25]:=
```
setMean = Mean[meanSet];
setStdev = StandardDeviation[meanSet];
setStdev2 = (Variance[meanSet])^0.5;
lowAge95 = setMean - 2 * setStdev;
highAge95 = setMean + 2 * setStdev;
lowAge90 = setMean - 1.645 * setStdev;
highAge90 = setMean + 1.645 * setStdev;
Print["mean=", setMean, " stdev=", setStdev];
Print["90% low=", lowAge90, " high=", highAge90];
Print["95% low=", lowAge95, " high=", highAge95];
```

```
mean=39.9991 stdev=3.61642

90% low=34.0501 high=45.9482

95% low=32.7663 high=47.232
```

But really, since we have less than 30 samples, slightly different math should be used. I used the 90% confidence level below. Note: In the blog entry, the "bet" used the high/low values above, not the ones below. The range values below are what I really should have used at the 95% level, but as you can see, the math is a little more complicated.

In[35]:=
```
setStdevP = (Length[meanSet] * Variance[meanSet] / (Length[meanSet] - 1))^0.5;
lowAgeP = setMean - 2 * setStdevP;
highAgeP = setMean + 2 * setStdevP;
Print["Pop STD: stdevP=", setStdevP, " lowP=", lowAgeP, " highP=", highAgeP];
```

```
Pop STD: stdevP=4.04328 lowP=31.9126 highP=48.0857
```

Now randomly create 10 groups of 5 DBAs each. Ask each group to determine their average age, write it donw on a pice of paper, and place the paper face down in front of me.

I now make the bet! "I will bet anyone $100 that 9 out of the 10 averages before me will between x and y." As I wrote in the blog, I actually used the 95% confidence numbers because of my small sample set.

Below are 10 groups of 5 ages each, along with their avereage and some other stats. I also included the code to quickly tell me how many of the avereage are within my selected range (using the 95% confidence level).

In[39]:=
```
lowAgeX = lowAge95;
highAgeX = highAge95;
groupSize = 5;
groupNum = 10;
yesCntrStd = 0;
noCntrStd = 0;
fullTestSet = {};
meanSet = {};
Table[
  testSet = fullSet[[i ;; i + groupSize - 1]];
  fullTestSet = Join[fullTestSet, testSet];
  testAvg = N[Mean[testSet]];
  testStdev = N[StandardDeviation[testSet]];
  testSE = testStdev / Length[testSet];
  AppendTo[meanSet, testAvg];

  If[(lowAgeX ≤ testAvg) && (highAgeX > testAvg),
   yesCntrStd = yesCntrStd + 1,
   noCntrStd = noCntrStd + 1
  ];
  Print[" avg=", testAvg, " stdev=", testStdev, " SE=", testSE, " set=", testSet];
  ,
  {i, 101, 100 + groupSize * groupNum, groupSize}
 ];
Print["Yes=", yesCntrStd, " No=", noCntrStd,
  " Yes%=", N[100 * yesCntrStd / (yesCntrStd + noCntrStd)]];
```

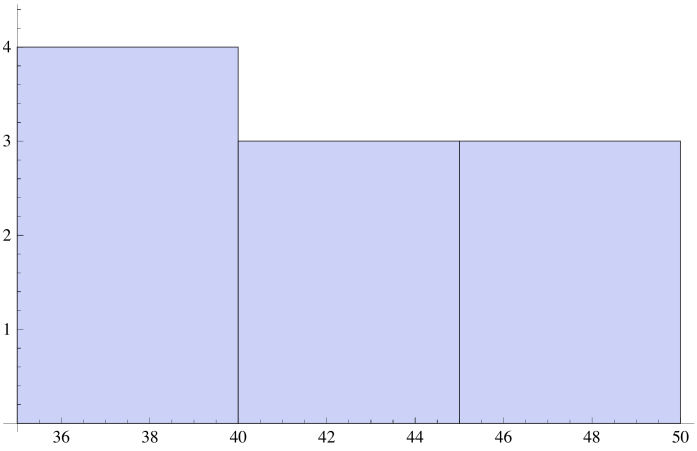
```
 avg=42.3903 stdev=8.6866 SE=1.73732 set={38.707, 33.6054, 42.1052, 56.8194, 40.7146}

 avg=35.8901 stdev=10.8646 SE=2.17292 set={48.6739, 27.2828, 46.7747, 29.5329, 27.1862}

 avg=46.1162 stdev=14.7221 SE=2.94442 set={62.3574, 30.5472, 58.3563, 47.6477, 31.6724}

 avg=45.2745 stdev=7.14389 SE=1.42878 set={54.662, 47.0582, 40.88, 47.826, 35.9465}

 avg=39.9453 stdev=5.83198 SE=1.1664 set={41.6942, 40.2792, 44.2132, 29.8962, 43.6436}

 avg=42.9772 stdev=10.5325 SE=2.10649 set={44.8206, 36.7735, 57.636, 29.6445, 46.0112}

 avg=39.3363 stdev=4.86551 SE=0.973103 set={37.1612, 34.0818, 36.6415, 45.7838, 43.0134}

 avg=35.5377 stdev=9.60032 SE=1.92006 set={36.0074, 25.208, 28.0371, 49.452, 38.9837}

 avg=42.1296 stdev=3.87174 SE=0.774348 set={38.7388, 48.5738, 41.1468, 39.711, 42.4779}

 avg=49.6692 stdev=5.92676 SE=1.18535 set={48.3371, 47.5775, 55.857, 41.4958, 55.0787}

Yes=9 No=1 Yes%=90.
```

One of the central limit theorem claims is a sample set made up of means from another sample set (like we have done) is normally distributed. Let's check it out! Because there are only 10 values the histogram is not convincing, but the noarmality test is! We want a pvalue > 0.05 and it is 0.98 which tells us the 10 values are very likely to be normally distributed!!

In[49]:=
```
meanSet
Histogram[meanSet]
DistributionFitTest[meanSet]
```

Out[49]= {42.3903, 35.8901, 46.1162, 45.2745, 39.9453, 42.9772, 39.3363, 35.5377, 42.1296, 49.6692}

Out[50]=



Out[51]= 0.978823

Below is ALL the data that was actually used in this test, including the aveage and a normality check. While I wanted the raw data to NOT be normal, statistically it is... even though it came from a Chi Square distribution!

In[52]:=
```
popSet = Join[set1, set2, set3, set4, set5, fullTestSet]
Length[popSet]
Mean[popSet]
DistributionFitTest[popSet]
Histogram[popSet]
```

Out[52]= {44.5757, 30.785, 35.6868, 29.9242, 50.1605, 26.0015, 30.1081, 34.0855, 43.95, 39.1439,
44.839, 33.9597, 41.7853, 46.8476, 38.4352, 36.8413, 57.4797, 45.0145, 48.7167,
31.2964, 38.2674, 26.8536, 58.0649, 51.6393, 35.5164, 38.707, 33.6054, 42.1052,
56.8194, 40.7146, 48.6739, 27.2828, 46.7747, 29.5329, 27.1862, 62.3574, 30.5472,
58.3563, 47.6477, 31.6724, 54.662, 47.0582, 40.88, 47.826, 35.9465, 41.6942, 40.2792,
44.2132, 29.8962, 43.6436, 44.8206, 36.7735, 57.636, 29.6445, 46.0112, 37.1612,
34.0818, 36.6415, 45.7838, 43.0134, 36.0074, 25.208, 28.0371, 49.452, 38.9837, 38.7388,
48.5738, 41.1468, 39.711, 42.4779, 48.3371, 47.5775, 55.857, 41.4958, 55.0787}

Out[53]= 75

Out[54]= 41.2841

Out[55]= 0.791982

Out[56]=