Workload: Mixed

Compare Sample Sets Statistical Analysis

version 1a, 26-Sep-2013
author: Craig Shallahamer, craig@orapub.com

Introduction

It's difficult for us humans to aborb and grasp more than just a few values. Statistics provides us with a way to grasp complexity by simplifying or abstracting. As Oracle performance specialist, we typically have a large number of numeric samples, comprising our sample set.

This notepad analyze two samples and then statisticall compares them.

Sample values

Enter your sample values; seperated by commas and the entire set enclosed in braces. Usually you can simply copy and paste the values directly into the notebook; which usually removes any line feeds or returns. It is common to not consider all sample vaues. Below you can specify threshholds for the minimum and maximum values to run through the analysis.

In[1]:=

```
cutoffMin = -1.0;
cutoffMax = 9999.0;
sampleSysstat = {481.85, 475.82, 476.88, 475.57, 482.67, 475.35,
   474.56, 480.53, 479.02, 487.38, 476.27, 474.79, 475.5, 479.35,
   476.14, 473.4, 474.12, 472.08, 478.1, 474.29, 480.26, 481.24, 477.31,
   476.68, 482.31, 475.61, 474.73, 483.72, 473.2, 473.66, 472.15};
sampleTimeModel = {526.17701, 511.961172, 517.199373, 522.195609,
   521.490723, 516.601465, 518.036242, 520.724838, 519.662995,
   528.518651, 518.117236, 519.045091, 515.726599, 516.331501,
   519.010098, 512.955017, 519.340048, 512.06815, 517.938263, 515.495629,
   522.695539, 523.968344, 518.9961, 519.867964, 518.858122, 518.570164,
   516.67345, 525.739073, 513.942872, 513.51393, 512.585072};
sampleSetName1 = "Sysstat";
sampleSetName2 = "Time Model";
sampleSetRaw1 = sampleSysstat;
sampleSetRaw2 = sampleTimeModel;
```

Basic Numeric Statistics

The basic numeric statistics shown are:

First five values simply lists the first five values of your sample set. Use this to check the data being used is what you think.
Number of samples is in fact the number of samples. Use this to check the data has been entered correctly.
Average is the statistical mean.
Median is the middle value after all the values are sorted. This is also the 50-percentile value.
Standard deviation is a measure of dispersion and is particularly valuable when the distribution is normal.
P-Value is a measure of contrast. In this instance, we are constrasting the sample set to the normal distribution. Loosely speaking, if the P-Value is greater than 0.05 then our sample set is likely to be normally distributed.

Sample Set #1

In[9]:=

```
sampleSetGood1 = {};
sampleSetBad1 = {};
Table[
  If[((sampleSetRaw1[[i]] ≤ cutoffMax) && (sampleSetRaw1[[i]] ≥ cutoffMin)),
   AppendTo[sampleSetGood1, sampleSetRaw1[[i]] ],
   AppendTo[sampleSetBad1, sampleSetRaw1[[i]] ]
  ]
  , {i, 1, Length[sampleSetRaw1]}
 ];

countRaw = Length[sampleSetRaw1];
countGood = Length[sampleSetGood1];
countBad = Length[sampleSetBad1];

firstFiveRaw = Take[sampleSetRaw1, 5];
firstFiveGood = Take[sampleSetGood1, 5];
firstFiveBad = Take[sampleSetBad1, 5];

avg = Round[N[Mean[sampleSetGood1]], 0.00010];
med = Round[N[Median[sampleSetGood1]], 0.00010];
std = Round[N[StandardDeviation[sampleSetGood1]], 0.0010];
pValue = Round[N[DistributionFitTest[sampleSetGood1]], 0.0000010];
pct90 = Round[N[Quantile[sampleSetGood1, 0.90]], 0.000010];
pct95 = Round[N[Quantile[sampleSetGood1, 0.95]], 0.000010];
pct99 = Round[N[Quantile[sampleSetGood1, 0.99]], 0.000010];
maxV = Max[sampleSetGood1];

Grid[{
  {"Statistics for Sample Set : " <> sampleSetName1 <> "\n"},
  {"Number of total samples", countRaw},
  {"Number of good samples", countGood},
  {"Number of bad samples", countBad},
  {"First five raw samples", firstFiveRaw},
  {"First five good samples", firstFiveGood},
  {"First five bad samples", firstFiveBad},
  {"Good Sample Details", "---"},
  {"  Average", avg},
  {"  Median (50%-tile)", med},
  {"  Maximum", maxV},
  {"  Percentiles (90,95,99)", {pct90, pct95, pct99}},
  {"  Standard deviation", std},
  {"  P-Value", pValue}
 },
 {Alignment → {Left},
  Frame → None}
]
```

Take::take : Cannot take positions 1 through 5 in {}. ≫

Out[26]=

```
Statistics for Sample Set : Sysstat

Number of total samples            31
Number of good samples             31
Number of bad samples              0
First five raw samples             {481.85, 475.82, 476.88, 475.57, 482.67}
First five good samples            {481.85, 475.82, 476.88, 475.57, 482.67}
First five bad samples             Take[{}, 5]
Good Sample Details                ---
  Average                          477.243
  Median (50%-tile)                476.14
  Maximum                          487.38
  Percentiles (90,95,99)           {482.31, 483.72, 487.38}
  Standard deviation               3.726
  P-Value                          0.026387
```

Sample Set #2

In[27]:=

```
sampleSetGood2 = {};
sampleSetBad2 = {};
Table[
  If[((sampleSetRaw2[[i]] ≤ cutoffMax) && (sampleSetRaw2[[i]] ≥ cutoffMin)),
   AppendTo[sampleSetGood2, sampleSetRaw2[[i]] ],
   AppendTo[sampleSetBad2, sampleSetRaw2[[i]] ]
  ]
  , {i, 1, Length[sampleSetRaw2]}
 ];

countRaw = Length[sampleSetRaw2];
countGood = Length[sampleSetGood2];
countBad = Length[sampleSetBad2];

firstFiveRaw = Take[sampleSetRaw2, 5];
firstFiveGood = Take[sampleSetGood2, 5];
firstFiveBad = Take[sampleSetBad2, 5];

avg = Round[N[Mean[sampleSetGood2]], 0.00010];
med = Round[N[Median[sampleSetGood2]], 0.00010];
std = Round[N[StandardDeviation[sampleSetGood2]], 0.0010];
pValue = Round[N[DistributionFitTest[sampleSetGood2]], 0.0000010];
pct90 = Round[N[Quantile[sampleSetGood2, 0.90]], 0.000010];
pct95 = Round[N[Quantile[sampleSetGood2, 0.95]], 0.000010];
pct99 = Round[N[Quantile[sampleSetGood2, 0.99]], 0.000010];
maxV = Max[sampleSetGood2];

Grid[{
  {"Statistics for Sample Set : " <> sampleSetName2 <> "\n"},
  {"Number of total samples", countRaw},
  {"Number of good samples", countGood},
  {"Number of bad samples", countBad},
  {"First five raw samples", firstFiveRaw},
  {"First five good samples", firstFiveGood},
  {"First five bad samples", firstFiveBad},
  {"Good Sample Details", "---"},
  {"  Average", avg},
  {"  Median (50%-tile)", med},
  {"  Maximum", maxV},
  {"  Percentiles (90,95,99)", {pct90, pct95, pct99}},
  {"  Standard deviation", std},
  {"  P-Value", pValue}
 },
 {Alignment → {Left},
  Frame → None}
]
```

Take::take : Cannot take positions 1 through 5 in {}. ≫

Out[44]=

```
Statistics for Sample Set : Time Model

Number of total samples                 31
Number of good samples                  31
Number of bad samples                   0
First five raw samples                  {526.177, 511.961,
                                         517.199, 522.196, 521.491}
First five good samples                 {526.177, 511.961,
                                         517.199, 522.196, 521.491}
First five bad samples                  Take[{}, 5]
Good Sample Details                     ---
  Average                               518.516
  Median (50%-tile)                     518.57
  Maximum                               528.519
  Percentiles (90,95,99)                {523.968, 526.177, 528.519}
  Standard deviation                    4.155
  P-Value                               0.530171
```

Basic Visual "Statistics"

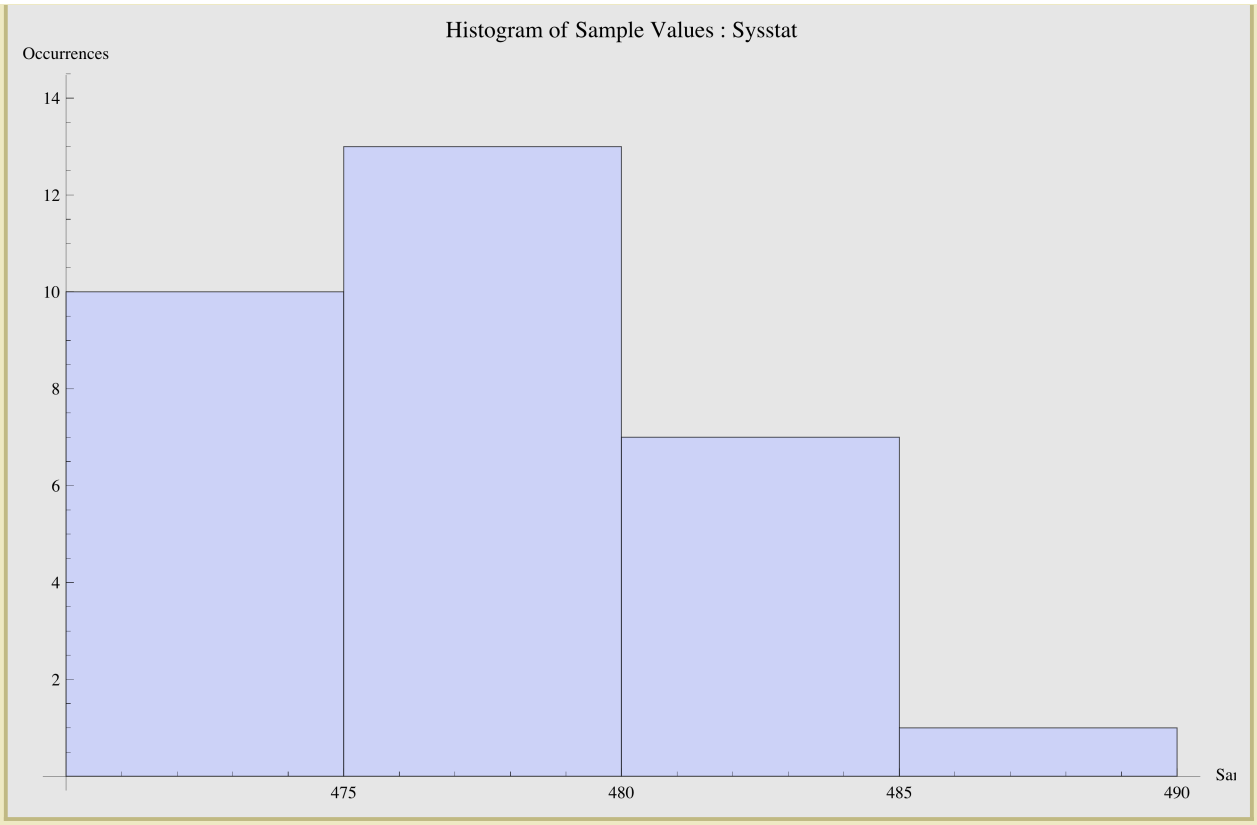
Histograms are a fantastic way to get a quick grasp of a large number of samples. Below are a select number of histogram, each focusing on a specific numeric quality.

In[45]:=
```
hLabel = "Sample Values";
vLabel = "Occurrences";

histStnd1 = Histogram[sampleSetGood1,
   PlotLabel → "Histogram of Sample Values : " <> sampleSetName1,
   AxesLabel → {hLabel, vLabel}];
histStndSmooth1 = SmoothHistogram[sampleSetGood1,
  PlotLabel → "Smoothed Histogram of Sample Values : " <> sampleSetName1 <>
    "\n(Probability Distribution Function)", AxesLabel → {hLabel, ""}];
histCC1 = Histogram[sampleSetGood1, Automatic, "CumulativeCount",
  PlotLabel → "Histogram of Sample Values : " <> sampleSetName1 <>
    "\nCumulative Count", AxesLabel → {hLabel, vLabel}];
histProb1 = Histogram[sampleSetGood1, Automatic, "Probability",
   PlotLabel → "Histogram of Sample Values : " <> sampleSetName1 <>
     "\nProbability", AxesLabel → {hLabel, "% Occurs"}];
histStndSmallBin1 = Histogram[sampleSetGood1, {0.250},
   PlotLabel → "Histogram of Sample Values : " <> sampleSetName1 <>
     "\nbin size 0.250", AxesLabel → {hLabel, vLabel}];

histStnd2 = Histogram[sampleSetGood2,
   PlotLabel → "Histogram of Sample Values : " <> sampleSetName2,
   AxesLabel → {hLabel, vLabel}];
histStndSmooth2 = SmoothHistogram[sampleSetGood2,
  PlotLabel → "Smoothed Histogram of Sample Values : " <> sampleSetName1 <>
    "\n(Probability Distribution Function)", AxesLabel → {hLabel, ""}];
histCC2 = Histogram[sampleSetGood2, Automatic, "CumulativeCount",
  PlotLabel → "Histogram of Sample Values : " <> sampleSetName2 <>
    "\nCumulative Count", AxesLabel → {hLabel, vLabel}];
histProb2 = Histogram[sampleSetGood2, Automatic, "Probability",
   PlotLabel → "Histogram of Sample Values : " <> sampleSetName2 <>
     "\nProbability", AxesLabel → {hLabel, "% Occurs"}];
histStndSmallBin2 = Histogram[sampleSetGood2, {0.250},
   PlotLabel → "Histogram of Sample Values : " <> sampleSetName2 <>
     "\nnbin size 0.250", AxesLabel → {hLabel, vLabel}];
Print[" "];
```

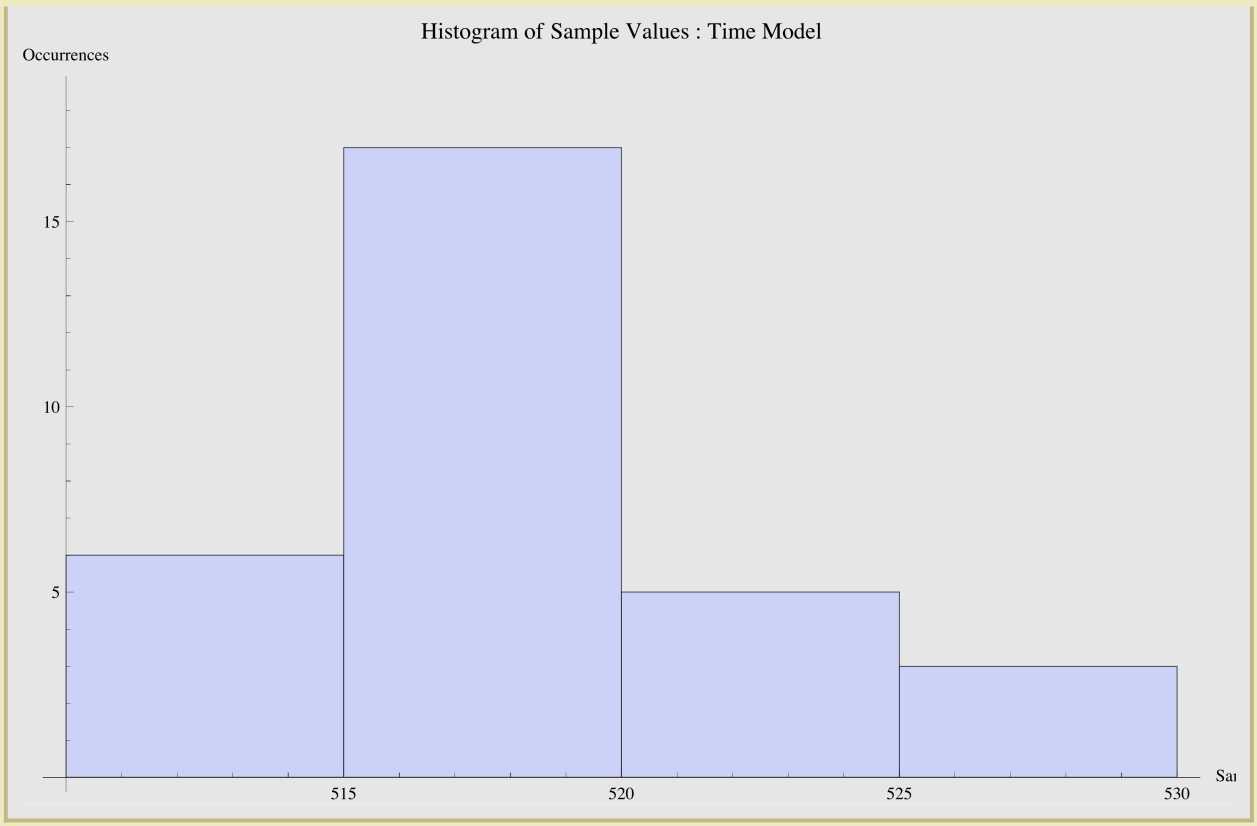
Below is a standard histogram, where each sample is shown as a single block placed on the vertical axis based on its value. Common sample values (i.e., blocks) show as high stacks.

In[56]:=
```
histStnd1
histStnd2
```

Out[56]=


Histogram of Sample Values : Sysstat
Occurrences
14
12
10
8
6
4
2
475
480
485
490
Sa

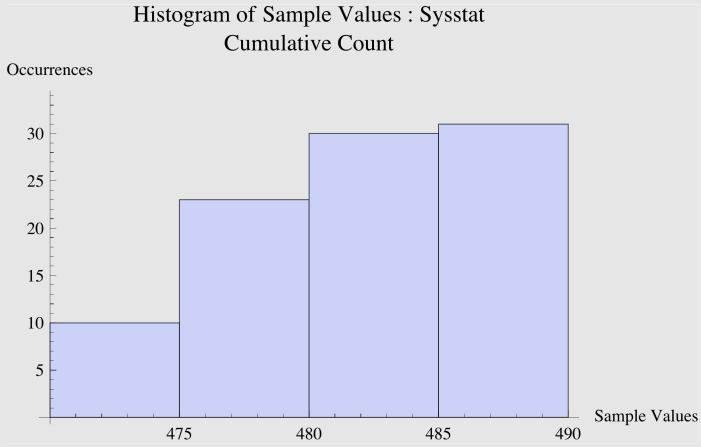
Out[57]=


Histogram of Sample Values : Time Model
Occurrences
15
10
5
515
520
525
530
Sa

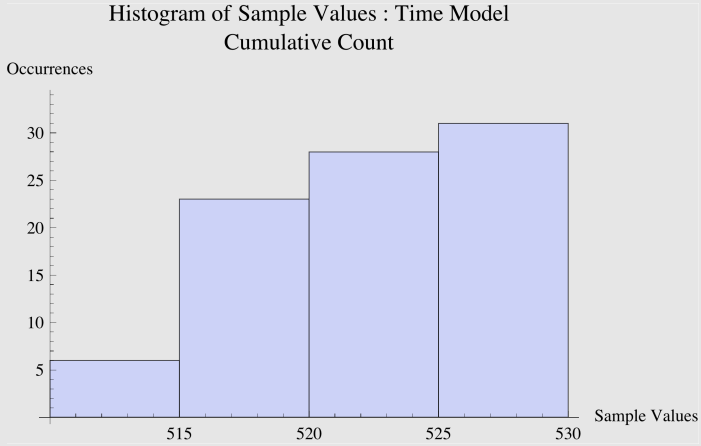
Below is known as a Cummlative Count histogram. Each vertical bar represents the total number of samples values that are less than or equal to the bin.

In[58]:=
```
histCC1
histCC2
```

Out[58]=



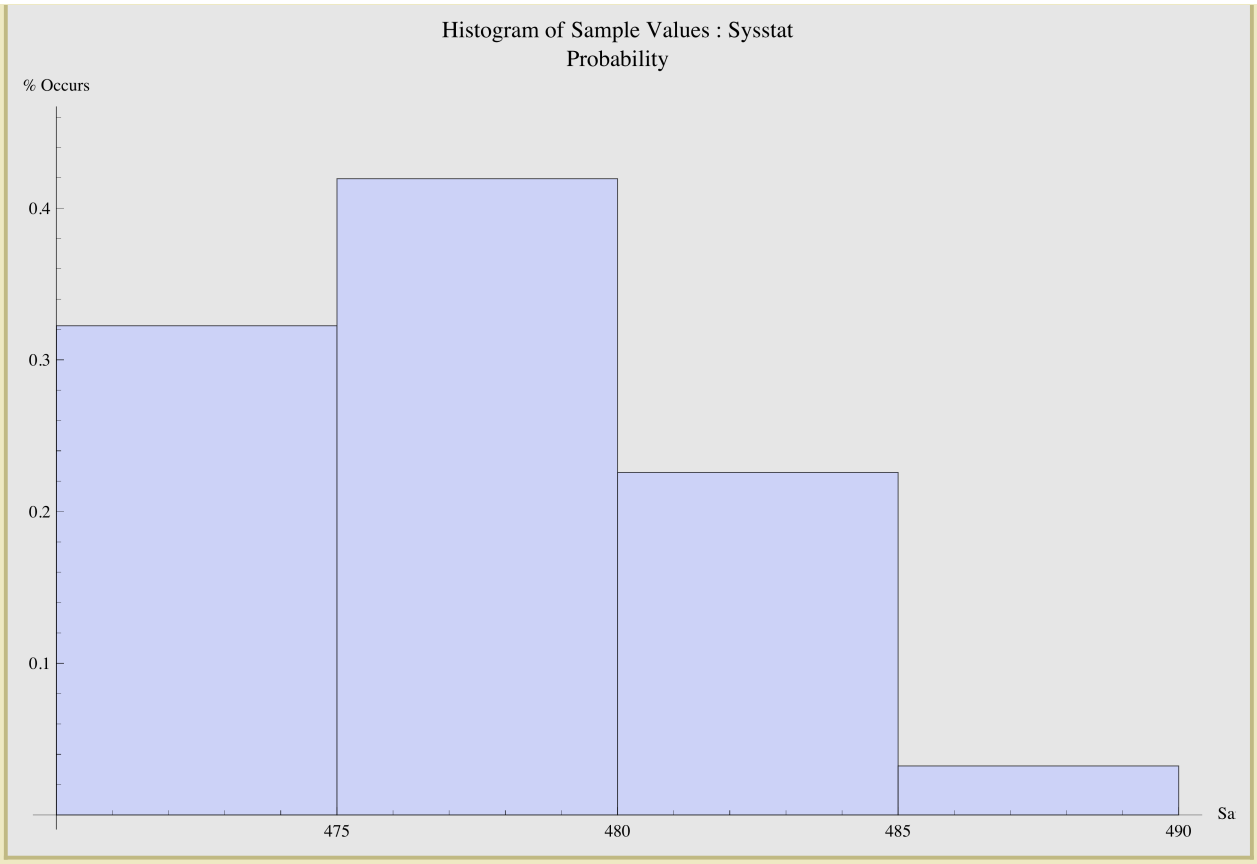
Out[59]=



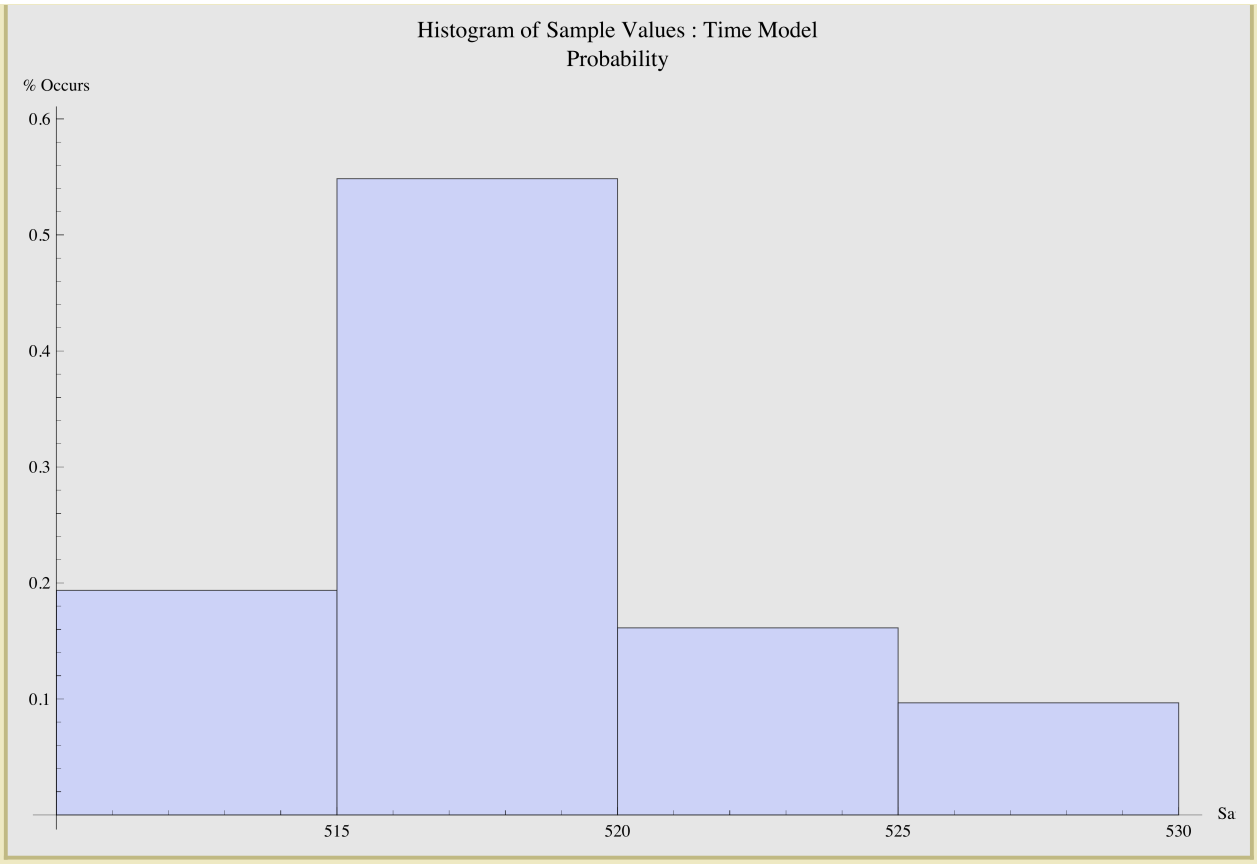
Below is a Probability Historgram. It will visually look exactly like the standard histogram but the vertical axis is a percentage value. Each vertical bar's height represents the percentage of values that is contains. In contrast, the standard histogram hight is the actual number of sample occurrences.

In[60]:=
```
histProb1
histProb2
```

Out[60]=


Histogram of Sample Values : Sysstat
Probability
% Occurs
0.4
0.3
0.2
0.1
475
480
485
490
Sa

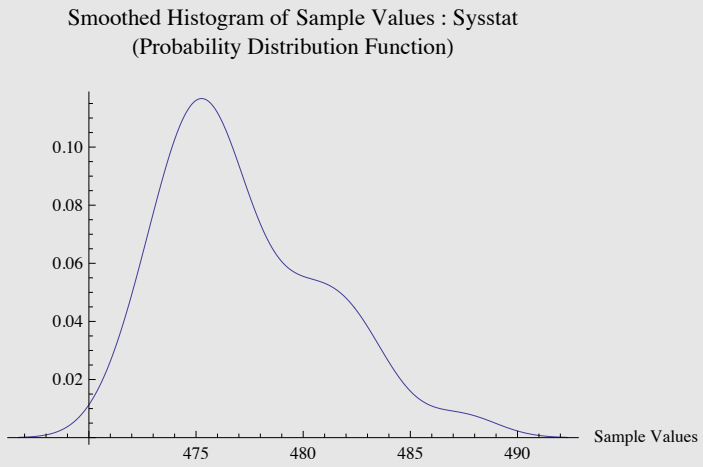
Out[61]=



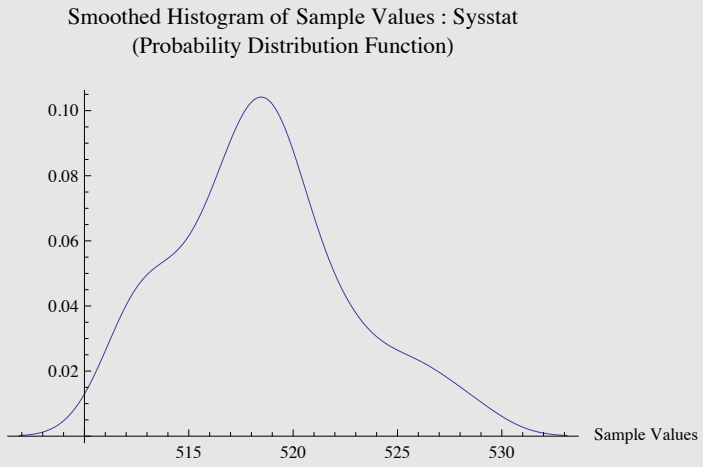
Below is a Smoothed Histogram. It will have a similar shape to the Standard Histogram, but will be mathematically smoothed. Sometimes this is a much more pleasent and informative visual, but not always. Remember, it is smoothed so it does not consist of the actual values. For example, you may see the line go negative, even though there are no negative values. In reality, the plot is the probability distribution function (PDF).

In[62]:=
```
histStndSmooth1
histStndSmooth2
```

Out[62]=



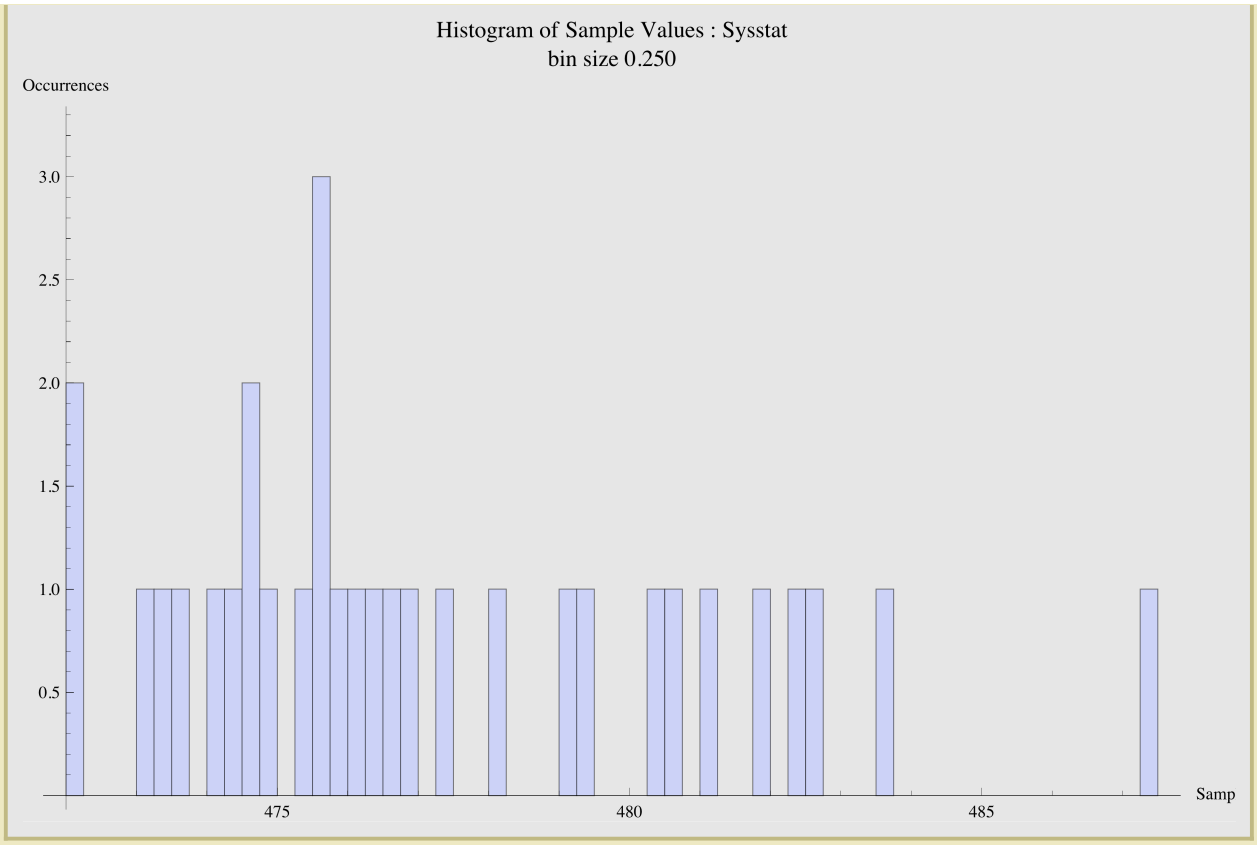
Out[63]=



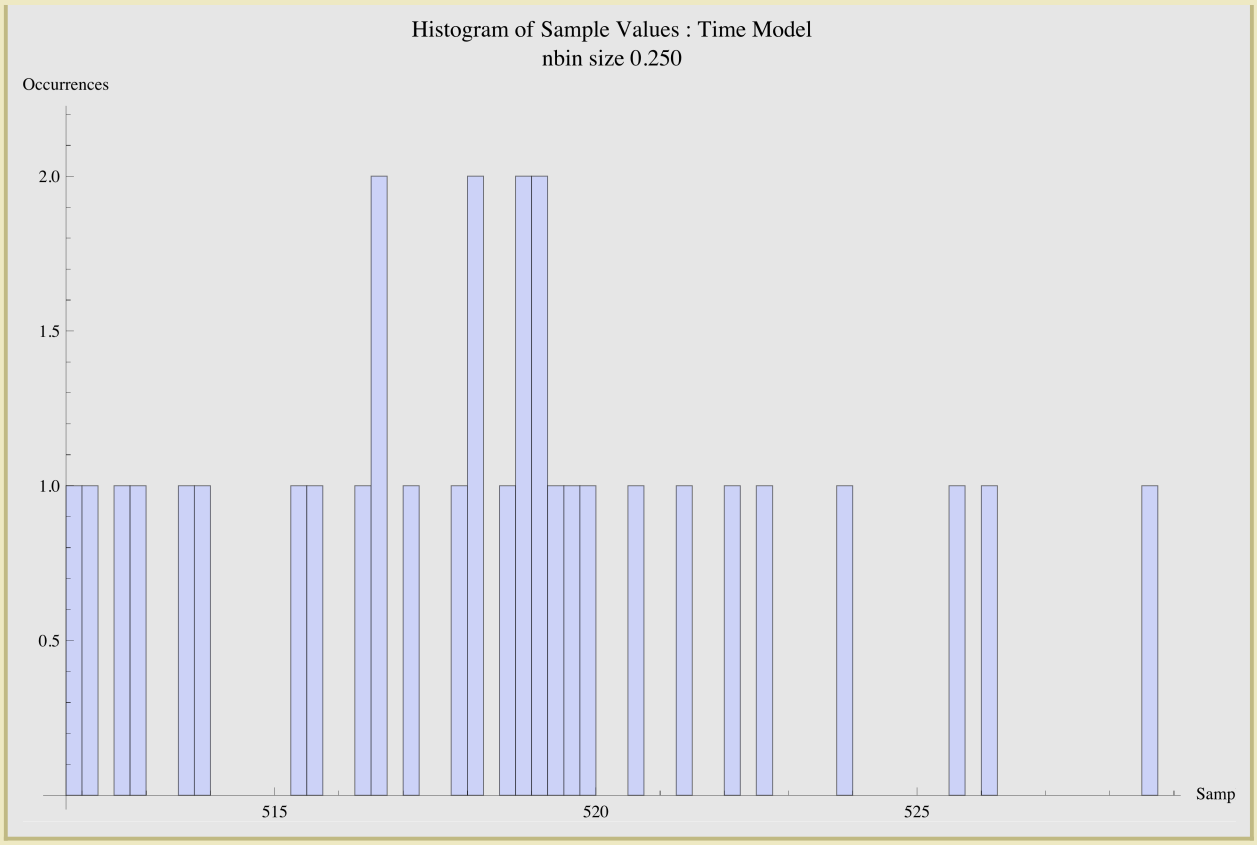
Below is a standard histogram but with the bin size set to 0.250. This is only useful when the sample values range below 1.0, such as when sampling SQL statements (we all hope). Sample sets with large sample values will likely not result in a plot.

In[64]:=
```
histStndSmallBin1
histStndSmallBin2
```

Out[64]=


Histogram of Sample Values : Sysstat
bin size 0.250
Occurrences
3.0
2.5
2.0
1.5
1.0
0.5
475
480
485
Samp

Out[65]=


Histogram of Sample Values : Time Model
nbin size 0.250
Occurrences
2.0
1.5
1.0
0.5
515
520
525
Samp

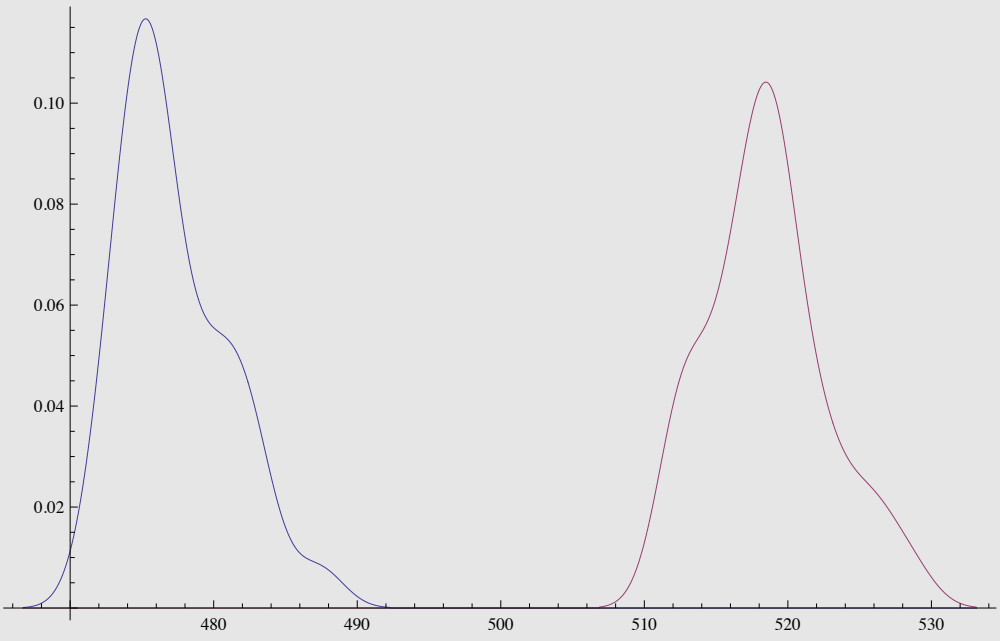
Compare the two sample sets

Below are the statisitcal tests to compare both numerically and visually the two sample sets. A simple t-test can be used IF both sample sets are normally distributed. (If their respective p-values are > 0.05 then we'll say they are normally distribugted.) If either sample set is not normally distributed we must use a location equivilency test. I'll let *Mathematica* pick location test to use.

I also included a straight difference calculation, in both value and percentage. Why? Just because there is a statistical difference does not mean we will feel or notice the difference. Knowing the difference can help us determine if a statistical difference makes a truly impactful difference.

In[66]:=
```
SmoothHistogram[{sampleSetGood1, sampleSetGood2}]
```

Out[66]=



In[67]:=
```
pValueTT = TTest[{sampleSetGood1, sampleSetGood2}]
```

Out[67]=
$$1.07469 \times 10^{-45}$$

In[68]:=
```
pValueLet = LocationEquivalenceTest[
  {sampleSetGood1, sampleSetGood2}, {"TestDataTable", "AutomaticTest"}]
```

Out[68]=

	Statistic	P-Value
K-Sample T	1695.44	1.07469×10^{-45}

, KSampleT

In[69]:=
```
medDiff = Abs[Median[sampleSetGood1] - Median[sampleSetGood2]]
medDiff / Median[sampleSetGood1]
medDiff / Median[sampleSetGood2]
```

Out[69]=
```
42.4302
```

Out[70]=
```
0.0891128
```

Out[71]=
```
0.0818215
```