CBC, Latches, and Response Time

Author: Craig Shallahamer (craig@orapub.com), Version 1d, 14-July-2011

Background and Purpose

The purpose of this notepad is to see if increasing CBC latches can decrease CPU and non-idle wait time per buffer get.

Experimental Data

Below is all the experimental data. The experiment was run on a Dell single four-core CPU, Oracle 11.2G. According to "c /proc/version": Linux version 2.6.18-164.el5PAE (mockbuild@ca-build10.us.oracle.com) (gcc version 4.1.2 20080704 (Red H 4.1.2-46)) #1 SMP Thu Sep 3 02:28:20 EDT 2009. There was a tremendous CBC latch contention load, the OS was CPU bott necked at between 95% and 100% utilization and the CPU run queue between 5 and 12. For each sample set, 60 samples we collected each with an interval of 180 seconds. There was also a calm time of 4 seconds between each sample. When t number of latches was changed, the instance was (obviously) cycled, the number of latched and chains checked, the identic CBC load program restarted, and the system was allowed to stabelize for three minutes before the sample collection began.

The order of sample data is sample number, elapsed time (seconds), buffer gets (session logical reads), instance non-idle w time (sec), instance CPU consumption (sec), and DB time (sec).

In[1]:=

```
ss256 = {1, 181.093777, 25 616 693, 707.2, 712.495681, 1550.711189, 2, 180.24696, 25 259 759, 718.52,
    707.889377, 1555.907435, 3, 180.605913, 25 387 380, 723.91, 712.916611, 1563.593177, 4, 180.739577,
    25 302 562, 718.67, 713.773486, 1565.880753, 5, 181.073032, 25 340 731, 727.01, 714.700339,
    1572.279433, 6, 180.891865, 25 187 066, 715.56, 711.233875, 1564.139884, 7, 180.303163, 25 235 271,
    717.68, 708.001367, 1555.93942, 8, 180.327352, 25 232 868, 717.6, 714.157421, 1564.679035, 9,
    181.135059, 25 171 962, 722.21, 712.916613, 1569.08229, 10, 180.749384, 25 158 450, 721.35, 710.299014,
    1561.75544, 11, 180.354793, 25 146 755, 731.4, 710.935912, 1570.422466, 12, 180.423949, 25 272 450,
    715.94, 713.707492, 1562.746141, 13, 181.27698, 25 161 275, 722.87, 716.690035, 1569.048431,
    14, 180.910418, 25 234 172, 722.98, 716.186117, 1573.586826, 15, 180.442985, 25 217 200, 722.26,
    709.556125, 1558.219369, 16, 180.453031, 25 136 001, 715.7, 715.865163, 1565.431488, 17, 180.400605,
    25 105 445, 716.68, 713.379542, 1562.766046, 18, 180.363533, 25 078 682, 730.02, 713.849473,
    1573.365168, 19, 181.134998, 25 090 193, 719.61, 715.169263, 1571.68373, 20, 180.771446, 25 109 736,
    724.06, 714.191418, 1570.855844, 21, 180.306655, 25 086 575, 716.3, 709.189186, 1550.76848, 22,
    180.776626, 25 078 173, 723.46, 711.928761, 1564.491226, 23, 180.376227, 25 116 693, 724.6, 709.787092,
    1561.862668, 24, 180.450843, 25 112 575, 717.32, 710.117036, 1558.673067, 25, 180.594034, 25 033 904,
    720.43, 713.475535, 1566.75662, 26, 180.493965, 25 078 837, 723.72, 711.057898, 1560.579815,
    27, 180.534611, 25 039 008, 718.94, 708.486281, 1555.09808, 28, 180.525511, 25 038 097, 728.73,
    713.997452, 1571.48528, 29, 180.868267, 25 053 064, 724.09, 717.289955, 1579.015314, 30, 180.726526,
    24 990 218, 728.04, 712.654654, 1568.994234, 31, 181.064011, 25 074 315, 720.76, 711.832778,
    1561.651717, 32, 180.842298, 25 005 329, 727.23, 714.609357, 1574.703246, 33, 181.394255, 25 035 936,
    717.73, 716.335097, 1572.712571, 34, 180.460659, 25 051 020, 722.77, 711.47683, 1559.401316,
    35, 180.570627, 25 039 514, 723.36, 711.984756, 1565.134863, 36, 180.427918, 24 989 826, 718.01,
    712.21372, 1555.630302, 37, 180.998524, 25 002 640, 728.68, 714.361392, 1575.462081, 38, 181.062487,
    25 007 644, 730.65, 716.829024, 1574.557354, 39, 180.934028, 25 005 890, 727.89, 712.457681,
    1569.249521, 40, 181.68984, 25 027 714, 729.5, 716.647043, 1580.497725, 41, 181.118936, 25 005 060,
    716.07, 714.425386, 1569.505682, 42, 180.238201, 25 014 183, 729.27, 709.391153, 1567.229449,
    43, 180.745539, 25 011 794, 720.2, 711.009901, 1557.012562, 44, 180.439792, 24 934 520, 733.73,
    712.064739, 1573.023074, 45, 180.417171, 25 004 055, 733.6, 707.99536, 1562.027266, 46, 180.446657,
    25 012 448, 719.2, 712.498677, 1564.886116, 47, 180.549129, 24 924 146, 722.85, 714.670351, 1567.92741,
    48, 181.238023, 24 968 204, 728.04, 714.444378, 1574.432192, 49, 181.225628, 24 929 578, 719.17,
    714.111438, 1567.244159, 50, 180.747706, 24 899 915, 731.48, 711.949759, 1579.042365, 51, 180.586662,
    24 960 509, 715.38, 714.767333, 1564.080677, 52, 180.452101, 24 897 610, 731.41, 713.134579,
    1573.639061, 53, 181.218539, 25 012 527, 719.81, 713.915463, 1561.028973, 54, 180.386978, 24 872 397,
    721.76, 709.876075, 1560.556377, 55, 180.670147, 24 993 120, 726.67, 711.108889, 1561.161963, 56,
    180.346618, 24 854 728, 735.93, 709.979059, 1575.679359, 57, 181.389562, 24 974 935, 726.8, 712.827631,
    1568.899526, 58, 180.855712, 24 850 801, 720.1, 712.005754, 1563.474324, 59, 180.786046, 24 923 707,
    735.81, 709.182183, 1568.680915, 60, 180.805406, 24 991 861, 731.37, 716.998998, 1579.18981};
ss512 = {1, 180.233063, 39 779 831, 454.43, 709.958059, 1364.234651, 2, 180.263832, 38 285 018, 478.07,
```

```
38 293 845, 479.54, 712.224715, 1395.112951, 5, 180.339543, 38 217 153, 481.25, 714.968301,
 1397.939564, 6, 180.338193, 38 153 874, 473.07, 716.177118, 1392.458655, 7, 181.323378, 38 053 395,
 475.87, 716.275104, 1400.867824, 8, 180.482484, 38 079 312, 490.14, 714.399386, 1409.831108, 9,
 180.559139, 37 963 737, 475.87, 718.143818, 1396.964276, 10, 180.229625, 37 962 643, 471.32, 710.550971,
 1382.137639, 11, 180.224537, 37 967 643, 485.12, 709.816081, 1396.612027, 12, 180.315353, 38 062 847,
 491.64, 713.834477, 1404.260518, 13, 180.463202, 38 012 250, 474.79, 715.409235, 1396.526995,
 14, 181.131785, 38 189 095, 475.63, 714.906309, 1395.623328, 15, 180.275086, 38 029 210, 483.91,
 710.707953, 1394.321784, 16, 180.940879, 38 006 493, 471.6, 715.927153, 1394.704402, 17, 181.01487,
 37 942 398, 481.79, 718.120821, 1407.713298, 18, 180.866583, 38 066 258, 469.19, 717.267954,
 1397.959286, 19, 180.323329, 38 078 626, 481.74, 711.470834, 1392.440422, 20, 180.706259, 37 995 711,
 481.34, 714.724333, 1397.557742, 21, 180.399353, 37 984 356, 477.88, 711.687812, 1386.71024, 22,
 180.68622, 37 921 962, 478.66, 715.26825, 1402.772734, 23, 180.716063, 38 065 041, 471.53, 715.199274,
 1389.023916, 24, 180.335274, 37 855 384, 476.82, 711.290859, 1392.013244, 25, 180.224958, 37 908 019,
 475.23, 709.877071, 1391.716468, 26, 180.734148, 37 997 775, 476.98, 714.199422, 1394.816732,
 27, 180.796058, 38 018 006, 467.62, 710.800939, 1376.857355, 28, 180.712536, 38 096 550, 485.27,
 715.289255, 1401.551934, 29, 180.498593, 37 900 926, 475.15, 714.12743, 1392.047058, 30, 180.416971,
 38 042 364, 486.97, 712.917615, 1392.213756, 31, 180.340371, 38 049 305, 469.93, 715.103273,
 1394.43341, 32, 180.458729, 38 071 866, 488.08, 716.685041, 1402.47642, 33, 180.222422, 37 846 863,
 472.73, 715.49922, 1391.545954, 34, 180.216384, 37 858 211, 475.23, 709.918073, 1388.967234,
 35, 181.142408, 37 933 529, 485.62, 719.622592, 1414.920925, 36, 180.614386, 37 779 994, 476.86,
 713.334549, 1392.772589, 37, 180.618187, 37 853 855, 473.47, 715.414236, 1397.025521, 38, 180.261776,
 37 866 309, 479.68, 716.789026, 1400.222894, 39, 180.897266, 37 995 511, 489.23, 714.310405,
 1400.836981, 40, 180.222029, 37 914 375, 473.14, 713.832473, 1392.761912, 41, 180.235059, 37 821 973,
 469.19, 713.286558, 1391.469702, 42, 181.134755, 37 767 491, 478.03, 717.989838, 1405.016423,
 43, 180.322757, 37 811 689, 481.45, 713.571518, 1397.092807, 44, 180.229962, 37 783 181, 489.84,
 715.014298, 1407.828917, 45, 180.31336, 37 830 132, 480.12, 716.196116, 1402.243978, 46, 180.259489,
 37 774 428, 470.83, 710.727943, 1387.514677, 47, 180.512944, 37 942 862, 479.99, 714.468379,
 1404.008384, 48, 180.269362, 37 940 420, 482.96, 713.578516, 1395.056871, 49, 180.322419, 37 848 657,
 474.09, 714.180418, 1391.899063, 50, 180.839975, 37 901 488, 480.28, 715.811174, 1399.56911, 51,
 180.258246, 37 794 157, 482.94, 711.698805, 1396.066035, 52, 181.01761, 38 130 278, 483.63, 714.406386,
 1396.148474, 53, 180.571228, 37 844 856, 484.83, 714.902311, 1395.308645, 54, 181.643591, 37 837 865,
 493.12, 716.463079, 1416.081375, 55, 180.447364, 37 910 446, 474.74, 716.227113, 1397.185238, 56,
 180.246418, 37 789 291, 467.61, 713.317553, 1384.04865, 57, 180.224919, 37 831 146, 479.99, 710.355,
 1389.815216, 58, 180.422642, 37 815 169, 490.68, 712.07574, 1399.473327, 59, 180.267498, 37 748 598,
 488.08, 714.974308, 1406.63575, 60, 180.324869, 37 793 252, 476.59, 712.611659, 1394.710783};
ss1024 = {1, 180.227828, 53 511 906, 255.55, 716.639049, 1219.189951, 2, 180.295481, 51 829 147,
 277.65, 707.085496, 1238.224708, 3, 180.294682, 51 951 582, 291.92, 702.776155, 1237.565301, 4,
 181.089126, 52 276 251, 257.34, 713.902461, 1226.619335, 5, 180.584455, 52 158 819, 287.12, 714.364394,
 1245.807111, 6, 180.501997, 52 488 981, 290.69, 714.933313, 1249.606903, 7, 180.67246, 52 472 769,
 263.66, 717.978844, 1235.642777, 8, 180.356081, 52 152 842, 285.81, 712.659657, 1242.887298, 9,
 180.218457, 52 016 935, 282.39, 714.495366, 1242.603546, 10, 180.305083, 52 538 163, 278.05, 714.343395,
 1234.816517, 11, 180.365859, 52 111 286, 282.45, 713.248563, 1238.590969, 12, 180.220939, 52 059 788,
 293.49, 709.563125, 1246.671279, 13, 180.402021, 52 100 434, 282.29, 711.682804, 1238.297702, 14,
 180.218969, 52 116 724, 292.24, 712.766631, 1250.747495, 15, 180.289304, 52 204 614, 288.2, 717.172965,
 1247.038756, 16, 181.115226, 52 318 112, 284.15, 717.821872, 1253.620634, 17, 180.886234, 52 324 903,
 286.23, 718.424777, 1256.288, 18, 180.501416, 51 255 012, 291.43, 703.174098, 1241.120701, 19,
 181.510528, 51 686 588, 286.25, 708.231327, 1246.059093, 20, 180.585591, 51 445 208, 283.6, 705.231784,
 1245.422872, 21, 180.993198, 51 728 769, 288.97, 712.377697, 1245.900229, 22, 180.516338, 51 885 737,
 273.99, 712.043747, 1233.995914, 23, 180.280839, 51 590 830, 253.95, 705.752704, 1214.866816,
 24, 180.360141, 51 323 478, 302.32, 703.835993, 1255.218292, 25, 180.653379, 51 016 098, 308.41,
 695.199304, 1257.672823, 26, 181.230848, 52 047 598, 292.01, 712.629659, 1250.979417, 27, 180.420856,
 51 829 394, 294.06, 711.106889, 1247.757158, 28, 180.561004, 51 985 783, 292.68, 714.231417,
 1253.849245, 29, 180.261731, 51 649 907, 295.24, 707.277473, 1242.050886, 30, 180.245699, 51 897 751,
 293.88, 712.797632, 1251.372773, 31, 180.654776, 52 291 327, 279.34, 716.161122, 1242.390026, 32,
 180.22165, 51 760 385, 286.53, 712.3547, 1247.284561, 33, 180.59866, 51 860 494, 247.18, 714.78733,
 1212.984237, 34, 180.221648, 51 484 486, 283.17, 710.984911, 1238.425757, 35, 180.30421, 51 450 240,
 277.33, 711.057892, 1245.320109, 36, 180.251284, 51 611 997, 268.58, 710.566971, 1230.892917,
 37, 180.649321, 51 788 738, 271.96, 713.60251, 1242.040383, 38, 181.164405, 51 671 885, 288.2,
 715.851166, 1252.854931, 39, 181.410594, 51 924 776, 286.68, 715.319248, 1252.818689, 40, 180.286354,
 51 843 528, 287.68, 710.993905, 1243.069986, 41, 180.445623, 50 744 522, 291.83, 699.201699,
 1249.551698, 42, 180.226128, 51 818 393, 251.33, 711.871774, 1219.936739, 43, 180.318359, 51 955 374,
 294.96, 713.566517, 1253.766454, 44, 180.218161, 52 205 533, 284.46, 715.839168, 1255.492034,
 45, 180.355895, 51 670 391, 277.35, 710.598971, 1232.624587, 46, 180.928183, 51 734 966, 289.18,
 710.313007, 1247.941596, 47, 180.248328, 51 825 440, 282.2, 707.951369, 1237.340588, 48, 181.906506,
 51 319 988, 289.89, 712.842625, 1257.103162, 49, 180.875368, 51 912 704, 282.53, 718.191812,
 1248.971564, 50, 180.254102, 51 789 207, 295.18, 713.903465, 1255.749247, 51, 182.414748, 52 335 278,
 291.38, 721.892251, 1263.872309, 52, 181.027473, 52 145 082, 300.47, 718.637743, 1262.942663,
 53, 180.714668, 51 832 110, 283.07, 718.220811, 1247.533837, 54, 181.050815, 51 850 253, 285.94,
 718.368787, 1251.98263, 55, 180.393616, 51 963 262, 278.23, 718.274794, 1243.96744, 56, 180.617253,
 51 897 672, 296.91, 716.091134, 1251.59526, 57, 180.576648, 51 870 355, 292.61, 717.121978,
 1255.674533, 58, 180.218878, 51 570 780, 283.96, 713.158576, 1240.872921, 59, 180.317543, 51 901 468,
 292.98, 711.638813, 1244.788293, 60, 180.588831, 51 676 381, 263.79, 708.106346, 1228.153002};
ss2048 = {1, 180.25324, 64 415 689, 116.01, 699.938583, 1088.331351, 2, 180.677454, 63 798 716, 104.94,
```

```
    1089.518503, 6, 180.377959, 63 724 194, 125.51, 704.829838, 1099.03548, 7, 180.428746, 63 452 186,
    122.32, 704.103954, 1098.827736, 8, 180.212122, 63 173 629, 123.08, 703.072111, 1102.507959, 9,
    180.213185, 63 707 261, 108.71, 708.952216, 1096.046367, 10, 181.264772, 62 677 351, 188.2, 712.167726,
    1161.850681, 11, 180.230775, 61 685 314, 291.02, 713.953459, 1186.969497, 12, 180.510684, 60 557 042,
    320.63, 702.782152, 1192.279756, 13, 180.348218, 61 323 405, 288.37, 712.887617, 1186.366951,
    14, 180.243004, 60 818 542, 312.89, 702.759158, 1191.055762, 15, 180.655557, 61 506 640, 295.09,
    711.885771, 1186.24254, 16, 180.304786, 61 151 583, 299.48, 710.381997, 1191.937151, 17, 180.657685,
    61 765 140, 283.94, 715.194273, 1184.69742, 18, 180.339888, 61 490 889, 292.37, 711.563819,
    1183.54961, 19, 180.769007, 59 974 194, 309.29, 711.208878, 1195.505264, 20, 180.229635, 59 123 423,
    307.63, 717.74288, 1210.622668, 21, 180.34817, 59 000 546, 306.95, 712.455682, 1197.397652, 22,
    180.39317, 58 898 086, 313.26, 713.313557, 1207.374656, 23, 180.850871, 58 960 104, 309.43, 714.53737,
    1200.56348, 24, 180.226265, 59 054 411, 303.14, 711.394842, 1193.782982, 25, 180.472308, 59 190 988,
    310.76, 716.213114, 1213.016216, 26, 180.516322, 59 120 751, 310.01, 716.453078, 1203.110223, 27,
    180.330106, 59 064 292, 306.54, 715.607203, 1202.145877, 28, 180.228721, 59 047 902, 309.9, 713.250561,
    1205.74163, 29, 180.218604, 58 956 637, 309.3, 711.994757, 1199.57604, 30, 180.269555, 58 398 819,
    308.05, 706.262631, 1197.213014, 31, 180.956597, 58 693 845, 311.34, 710.071045, 1197.012035,
    32, 180.745797, 59 023 043, 300.86, 714.593361, 1202.062291, 33, 180.312287, 59 126 031, 304.64,
    713.318552, 1196.665654, 34, 180.214622, 59 290 946, 304.56, 712.892624, 1195.850981, 35, 180.229548,
    58 927 025, 310.26, 717.630894, 1209.874739, 36, 180.240638, 59 020 185, 302.22, 712.847627,
    1195.556872, 37, 180.201784, 58 802 108, 314.03, 712.534672, 1206.180818, 38, 180.311521, 59 068 153,
    309.14, 716.385088, 1204.673334, 39, 180.347255, 59 043 657, 299.33, 711.878773, 1191.778239, 40,
    180.231803, 58 865 529, 305.35, 711.0389, 1196.964481, 41, 180.642941, 58 771 608, 312.45, 715.04029,
    1213.529323, 42, 180.294671, 58 854 350, 312.7, 709.556125, 1198.552209, 43, 180.54439, 58 913 513,
    300.37, 714.226414, 1199.052137, 44, 180.747228, 58 983 156, 309.57, 709.507134, 1194.386546,
    45, 180.331381, 59 036 979, 313.86, 710.976911, 1201.323593, 46, 180.249918, 59 021 772, 311.13,
    712.163732, 1197.533189, 47, 180.469864, 58 946 912, 314.16, 718.435768, 1214.312414, 48, 180.353224,
    59 009 564, 310.21, 712.63166, 1205.406737, 49, 181.161106, 58 846 982, 308.99, 717.343942,
    1205.928331, 50, 180.279246, 58 677 871, 304.75, 716.176117, 1202.678903, 51, 180.220738, 58 849 304,
    313.79, 712.708647, 1205.55396, 52, 180.883722, 58 848 509, 315.09, 714.669347, 1208.130469,
    53, 180.485361, 58 783 433, 313.76, 710.124043, 1199.844518, 54, 180.239983, 58 627 165, 309.98,
    707.533427, 1203.458916, 55, 180.249819, 58 745 042, 314.35, 709.742096, 1204.36485, 56, 180.337676,
    58 941 001, 307.54, 715.347246, 1202.712831, 57, 180.226582, 58 821 309, 310.55, 712.452687,
    1203.498691, 58, 180.478155, 58 427 190, 319.66, 711.607814, 1208.383657, 59, 180.346022, 58 728 118,
    305.71, 710.404998, 1195.714675, 60, 180.907704, 58 750 018, 314.36, 716.455075, 1207.958267};
ss4096 = {1, 180.232239, 70 571 051, 82.2, 705.472742, 1060.38432, 2, 180.487527, 70 174 358, 84.76,
    697.244, 1061.324129, 3, 180.329784, 68 885 933, 92.92, 686.913569, 1065.33443, 4, 180.212066,
    69 583 372, 89.43, 694.795369, 1066.214295, 5, 180.235112, 69 226 302, 96.29, 689.130225, 1060.873715, 6,
    180.228839, 70 190 030, 100.44, 702.827147, 1066.973286, 7, 180.344807, 69 541 549, 86.83, 698.518807,
    1070.585292, 8, 180.224323, 69 231 665, 88.68, 691.190919, 1069.893493, 9, 180.225351, 70 065 589,
    92.64, 698.270842, 1061.206138, 10, 180.452496, 69 767 005, 91.62, 701.605336, 1072.703582, 11,
    180.217513, 69 908 726, 91.87, 696.607092, 1064.005011, 12, 180.243545, 71 058 509, 78.46, 714.196417,
    1070.872383, 13, 180.228619, 69 708 450, 90.48, 691.556857, 1054.560259, 14, 180.213049, 69 200 984,
    102.09, 692.09078, 1068.878659, 15, 180.38508, 69 410 476, 97.01, 693.047635, 1068.785337, 16,
    180.306674, 69 813 934, 86.27, 698.807764, 1071.880178, 17, 180.226614, 69 616 131, 93.24, 695.601248,
    1063.567817, 18, 180.228583, 70 973 302, 78.41, 706.134646, 1061.024904, 19, 180.32175, 69 591 098,
    100.1, 697.088017, 1076.156751, 20, 180.229203, 69 857 640, 92.62, 698.365833, 1061.682449, 21,
    180.342839, 70 252 778, 96.14, 698.468808, 1070.108779, 22, 180.28683, 70 359 972, 91.08, 705.864687,
    1069.218363, 23, 180.218131, 69 787 290, 91.12, 698.964738, 1067.034371, 24, 180.203648, 70 236 254,
    87.67, 706.354614, 1066.26405, 25, 180.22997, 69 875 574, 92.29, 697.08602, 1062.736081, 26,
    180.229967, 69 475 423, 86.84, 691.600852, 1065.683106, 27, 180.229141, 69 777 598, 91.09, 697.459966,
    1063.788653, 28, 180.289519, 69 652 673, 90.94, 696.773071, 1061.841847, 29, 180.411766, 69 541 311,
    92.7, 694.749376, 1066.144783, 30, 180.401719, 69 425 506, 89.1, 692.477717, 1064.697137, 31,
    180.223917, 69 947 862, 88.69, 697.877909, 1064.775108, 32, 180.230879, 69 622 436, 91.63, 696.837054,
    1062.46581, 33, 180.227068, 68 444 829, 96.13, 681.618376, 1062.719365, 34, 180.212391, 69 259 164,
    89.41, 690.703988, 1066.796985, 35, 180.239001, 69 806 767, 95.87, 698.199851, 1059.396323, 36,
    180.203333, 70 185 562, 97.65, 704.531889, 1070.207848, 37, 180.23123, 71 079 674, 76.19, 704.63487,
    1052.771673, 38, 180.382185, 69 581 311, 97, 695.267297, 1071.457738, 39, 180.24282, 69 870 989,
    92.99, 700.908437, 1071.691932, 40, 180.224314, 69 067 276, 104.87, 685.901719, 1068.677237, 41,
    180.221019, 68 701 147, 99.17, 684.715903, 1060.496365, 42, 180.219736, 69 836 539, 83.92, 699.806611,
    1067.943415, 43, 180.222515, 70 554 513, 79.96, 703.47805, 1055.975699, 44, 180.214606, 69 966 144,
    91.39, 698.389821, 1064.590124, 45, 180.252483, 70 109 119, 94.17, 701.572338, 1067.217335, 46,
    180.214282, 70 072 969, 94.52, 699.2127, 1061.10434, 47, 180.20621, 69 335 277, 87.54, 696.326133,
    1076.201408, 48, 180.230711, 69 670 671, 92.51, 696.287149, 1062.378861, 49, 180.228043, 70 719 923,
    75.72, 707.678411, 1060.519277, 50, 180.329342, 69 518 737, 96.89, 696.976035, 1074.323707, 51,
    180.225106, 70 604 434, 87.67, 705.06881, 1062.383584, 52, 180.223873, 70 233 642, 82.47, 702.862144,
    1063.800171, 53, 180.21747, 69 353 400, 99.25, 696.212155, 1077.01888, 54, 180.484671, 70 245 466,
    89.56, 702.923127, 1072.131412, 55, 180.221075, 70 266 249, 80.99, 703.227089, 1062.9361, 56,
    180.331909, 70 221 274, 83.2, 703.263078, 1066.056114, 57, 180.224729, 69 451 729, 87.58, 696.242154,
    1069.206024, 58, 180.364813, 68 513 708, 93.22, 685.687756, 1070.631064, 59, 181.065297, 69 278 800,
    94.23, 696.248148, 1073.042319, 60, 180.325776, 69 336 384, 96.94, 694.584399, 1071.274958};
ss8192 = {1, 180.217151, 74 774 687, 62.44, 703.468057, 1042.35379, 2, 180.327481, 74 174 241, 72.21,
    699.154709, 1041.156993, 3, 180.211468, 73 783 895, 75.89, 690.856971, 1040.773831, 4, 180.222323,
    73 859 573, 67.72, 695.622242, 1046.461231, 5, 180.245719, 74 516 434, 76.15, 697.575945, 1037.286037,
```

```
    74.46, 680.954474, 1052.368837, 10, 180.203466, 73 967 835, 73.06, 692.652698, 1046.327069, 11,
    180.204908, 74 164 293, 66.84, 695.638241, 1043.243571, 12, 180.214893, 74 318 145, 75.09, 698.745767,
    1046.662023, 13, 180.2238, 72 707 364, 80.44, 683.610065, 1054.122634, 14, 180.224768, 74 563 980,
    65.38, 703.953978, 1043.167261, 15, 180.225059, 75 515 564, 59.04, 711.698799, 1041.186799, 16,
    180.284632, 74 144 536, 67.32, 695.557255, 1049.351286, 17, 180.212262, 73 964 584, 78.1, 692.572704,
    1047.338677, 18, 180.209348, 74 515 923, 70.72, 699.017724, 1039.855161, 19, 180.2346, 72 836 749,
    77.7, 683.726052, 1052.414187, 20, 180.21427, 73 917 925, 67.08, 694.99134, 1053.007201, 21,
    180.301481, 72 882 492, 82.81, 685.369797, 1054.881146, 22, 180.221159, 72 727 369, 83.77, 684.793889,
    1055.981079, 23, 180.304452, 73 604 513, 68.83, 690.971953, 1040.787689, 24, 180.222958, 73 926 558,
    73.94, 696.607095, 1048.05399, 25, 180.225691, 74 420 208, 66.13, 702.176251, 1039.300812, 26,
    180.524429, 74 385 548, 58.93, 701.658326, 1040.75932, 27, 180.21795, 73 353 098, 69.22, 690.918962,
    1048.885365, 28, 180.225156, 73 492 896, 71.5, 687.691448, 1048.527491, 29, 180.225978, 73 031 910,
    80.51, 685.638763, 1047.731584, 30, 180.22169, 74 920 311, 59.67, 707.694406, 1044.255024, 31,
    180.204024, 75 021 629, 65.01, 703.690017, 1039.734588, 32, 180.218879, 73 871 138, 63.41, 697.586947,
    1046.540396, 33, 180.199581, 73 992 696, 66.8, 694.215459, 1048.441411, 34, 180.204547, 75 091 812,
    59.63, 706.405607, 1051.645423, 35, 180.215656, 73 089 658, 75.38, 688.202372, 1049.188833, 36,
    180.214536, 73 606 667, 76.2, 693.457575, 1043.807614, 37, 180.227827, 74 089 409, 63.76, 696.130164,
    1040.435586, 38, 180.288328, 72 132 951, 77.43, 675.069374, 1044.111584, 39, 180.223997, 73 685 486,
    74.54, 691.183914, 1041.705398, 40, 180.540022, 74 030 694, 72.04, 697.195, 1048.246972, 41,
    180.214898, 73 450 822, 72.37, 689.067237, 1045.821339, 42, 180.298144, 74 250 910, 62.07, 700.253541,
    1049.841388, 43, 180.203861, 74 625 508, 69.59, 704.349916, 1049.366535, 44, 180.324255, 74 381 608,
    70.75, 700.140557, 1052.036298, 45, 180.421725, 73 634 208, 76.95, 689.358193, 1049.910767, 46,
    180.227527, 73 883 141, 75.07, 694.35444, 1050.662826, 47, 180.387704, 73 789 272, 75.74, 691.395885,
    1049.925526, 48, 180.272468, 74 896 511, 71.96, 705.726709, 1048.157077, 49, 180.216496, 74 652 629,
    68.89, 702.213238, 1043.864827, 50, 180.242258, 72 936 229, 76.37, 687.022553, 1057.02407, 51,
    180.21773, 73 795 957, 66.69, 693.34559, 1043.471194, 52, 180.579614, 73 651 964, 69.86, 691.450878,
    1035.464318, 53, 180.452077, 73 137 216, 67.9, 687.856421, 1050.241923, 54, 180.221084, 74 909 478,
    65.13, 706.314616, 1039.305061, 55, 180.402252, 73 596 745, 70.42, 694.168462, 1042.749185, 56,
    180.281802, 74 344 658, 66.27, 699.915595, 1048.477743, 57, 180.216348, 73 275 708, 80.62, 685.698749,
    1047.813281, 58, 180.218054, 72 994 584, 73.85, 689.825123, 1050.728691, 59, 180.210265, 75 073 286,
    58.17, 706.591575, 1039.044778, 60, 180.314091, 72 313 566, 77.96, 682.558231, 1053.498812};
ss16384 = {1, 180.215349, 77 373 280, 51.71, 686.631612, 1027.614296, 2, 180.380316, 77 163 621, 58.7,
    685.980705, 1034.444526, 3, 180.228241, 77 106 547, 55.13, 694.413423, 1030.652294, 4, 180.218476,
    77 589 190, 59.65, 698.492806, 1034.387377, 5, 180.218211, 77 547 566, 55.04, 691.718838, 1030.996367,
    6, 180.263145, 77 440 897, 57.27, 695.157311, 1026.579029, 7, 180.221478, 77 638 335, 53.98, 698.992734,
    1031.320214, 8, 180.226113, 75 559 455, 59.94, 678.069909, 1034.388932, 9, 180.207072, 75 509 464,
    64.8, 675.814259, 1031.111441, 10, 180.213587, 77 692 943, 47.8, 699.549647, 1030.125074, 11,
    180.22802, 76 360 880, 53.62, 684.803883, 1028.262093, 12, 180.227229, 76 755 096, 55.15, 690.798978,
    1042.708191, 13, 180.221887, 76 968 092, 58.12, 692.632698, 1028.900682, 14, 180.20058, 76 662 995,
    57.47, 687.965411, 1031.451133, 15, 180.232063, 78 086 796, 51.03, 703.95198, 1032.452736, 16,
    180.216498, 78 649 268, 44.73, 708.203331, 1028.68676, 17, 180.225004, 76 658 709, 56.78, 691.3059,
    1025.562781, 18, 180.223577, 75 760 112, 60.76, 677.835952, 1037.063299, 19, 180.274714, 77 390 353,
    50.46, 700.089567, 1032.160323, 20, 180.283951, 76 428 450, 58.78, 678.426854, 1026.658838, 21,
    180.225642, 76 569 214, 60.85, 690.292051, 1034.803041, 22, 180.22199, 76 902 324, 62.89, 689.578165,
    1031.263339, 23, 180.289481, 77 602 953, 46.16, 697.368979, 1035.291133, 24, 180.386353, 77 397 277,
    59.67, 694.40643, 1034.611434, 25, 180.397675, 77 467 412, 54.1, 699.219699, 1030.770778, 26,
    180.210083, 77 685 151, 51.67, 699.020723, 1028.659556, 27, 180.223676, 76 650 738, 55.82, 687.615462,
    1030.89156, 28, 180.294165, 76 662 569, 63.31, 690.43803, 1033.575687, 29, 180.226663, 77 619 526,
    53.97, 698.37383, 1025.209486, 30, 180.207361, 76 623 068, 62.22, 687.912413, 1029.555508, 31,
    180.231645, 77 568 392, 50.71, 699.564643, 1033.695151, 32, 180.348731, 76 730 859, 57.64, 685.663756,
    1024.170456, 33, 180.2014, 76 140 411, 58.87, 682.693206, 1028.58034, 34, 180.228987, 77 179 820,
    57.58, 692.250759, 1034.32331, 35, 180.419846, 77 922 214, 50.53, 701.127409, 1026.124801, 36,
    180.22072, 76 934 709, 53.81, 694.032485, 1040.112003, 37, 180.257346, 77 353 474, 55.28, 692.257754,
    1031.914027, 38, 180.223291, 76 717 985, 61.49, 681.294422, 1035.939663, 39, 180.227064, 75 855 291,
    60.83, 679.969624, 1037.46414, 40, 180.321929, 75 345 971, 61.28, 675.600289, 1031.36393, 41,
    180.229913, 77 168 746, 60.34, 690.416034, 1029.205032, 42, 180.29841, 76 614 313, 57.51, 689.298204,
    1037.248979, 43, 180.220131, 77 092 195, 53.71, 692.546709, 1031.651765, 44, 180.212871, 77 495 174,
    57.26, 692.98064, 1028.855828, 45, 180.204962, 78 231 167, 56.18, 701.097414, 1026.771544, 46,
    180.223032, 76 680 202, 60.48, 687.565468, 1027.282554, 47, 180.248907, 75 963 415, 61.7, 682.764201,
    1040.598132, 48, 180.225463, 76 177 329, 55.37, 679.518693, 1022.885569, 49, 180.228136, 77 300 473,
    58.7, 695.465266, 1035.219187, 50, 180.219599, 77 719 143, 48.79, 697.809909, 1024.707293, 51,
    180.274418, 78 135 391, 47.7, 701.229391, 1027.555441, 52, 180.22831, 78 278 027, 48.37, 703.299077,
    1021.536337, 53, 180.264032, 76 880 751, 58.6, 684.822887, 1031.19264, 54, 180.227736, 76 079 351,
    57.28, 683.294117, 1030.918071, 55, 180.213643, 76 227 489, 56.72, 687.03455, 1034.108107, 56,
    180.222601, 77 138 147, 58.94, 692.588703, 1023.617569, 57, 180.217255, 76 942 224, 54.38, 693.84352,
    1034.432816, 58, 180.162974, 76 201 005, 59.86, 683.485085, 1028.318376, 59, 180.228363, 77 394 448,
    53.9, 695.016337, 1028.8085, 60, 180.228755, 75 681 642, 62.18, 683.560075, 1042.166898};
ss32768 = {1, 180.221966, 79 506 939, 42.28, 694.707381, 1026.291497, 2, 180.211351, 78 906 286, 40.69,
    692.206763, 1023.600727, 3, 180.219132, 79 420 875, 42.04, 696.317136, 1019.857473, 4, 180.212147,
    79 990 447, 40.34, 699.910594, 1010.762685, 5, 180.218121, 79 136 233, 43.72, 692.216759, 1019.455177,
    6, 180.215492, 78 783 081, 41.25, 690.470027, 1023.058538, 7, 180.230688, 78 080 951, 41.57, 679.200738,
    1019.53868, 8, 180.345352, 79 881 493, 42.55, 699.955584, 1027.648981, 9, 180.218346, 79 808 867,
    39.64, 702.665171, 1022.395812, 10, 180.293365, 78 477 958, 46.92, 688.770283, 1027.856817, 11,
```

```
    39.26, 699.201694, 1025.220706, 15, 180.228393, 79 603 452, 41.45, 697.802914, 1029.293958, 16,
    180.295237, 79 225 056, 40.79, 694.322437, 1019.941203, 17, 180.225775, 78 911 364, 48.76, 694.90135,
    1025.644424, 18, 180.224265, 80 181 088, 35.02, 703.070113, 1014.655306, 19, 180.286651, 78 318 355,
    48.2, 684.483939, 1027.240984, 20, 180.225806, 79 963 006, 36.02, 702.684166, 1021.267471, 21,
    180.198353, 79 499 380, 39.17, 695.396281, 1015.472769, 22, 180.215055, 79 423 434, 41.59, 694.80137,
    1016.511551, 23, 180.203219, 78 177 286, 48.56, 685.933713, 1022.891479, 24, 180.265777, 78 348 334,
    46.46, 685.813733, 1021.282022, 25, 180.233047, 77 859 326, 53.96, 679.940633, 1020.827537, 26,
    180.208007, 78 318 995, 40.08, 686.504623, 1023.08827, 27, 180.267015, 79 840 537, 37.5, 696.902053,
    1017.34836, 28, 180.345499, 78 842 213, 42.11, 689.10023, 1024.845566, 29, 180.211357, 79 040 430,
    42.1, 693.476567, 1021.92709, 30, 180.229785, 77 820 566, 47.57, 680.120598, 1016.550072, 31,
    180.228556, 78 056 777, 44.44, 678.480851, 1015.013476, 32, 180.250292, 78 561 598, 40.95, 687.746443,
    1019.610926, 33, 180.160254, 78 173 122, 42.57, 679.099749, 1029.776949, 34, 180.202751, 79 372 434,
    44.45, 695.743225, 1035.079229, 35, 180.254637, 79 615 557, 45.75, 699.275686, 1018.333602, 36,
    180.223347, 78 235 862, 44.95, 682.153291, 1018.761995, 37, 180.259318, 80 414 661, 40.4, 706.368607,
    1021.132041, 38, 180.241543, 80 438 323, 37.05, 705.170784, 1012.170276, 39, 180.22492, 78 067 232,
    43.03, 684.265968, 1024.77678, 40, 180.236676, 78 472 277, 45.06, 688.709297, 1026.893231, 41,
    180.208485, 79 903 193, 38.59, 702.457203, 1023.952371, 42, 180.222234, 80 861 038, 32.08, 708.035355,
    1012.606193, 43, 180.23797, 80 956 254, 34.41, 711.187875, 1011.929635, 44, 180.298482, 80 835 051,
    32.32, 713.420536, 1016.865578, 45, 180.209666, 80 722 732, 35.63, 708.260325, 1013.678594, 46,
    180.340196, 78 396 218, 45.16, 683.361105, 1014.002001, 47, 180.213429, 79 314 597, 39.29, 696.362135,
    1019.634859, 48, 180.276005, 80 177 430, 37.49, 702.939126, 1013.308371, 49, 180.226896, 78 625 799,
    47.68, 689.476176, 1020.089108, 50, 180.230415, 78 380 049, 43.85, 686.830579, 1020.124858, 51,
    180.2215, 79 438 670, 35.06, 695.761222, 1024.961721, 52, 180.255528, 78 337 693, 47, 684.043004,
    1027.49119, 53, 180.206395, 80 473 554, 34.08, 705.372761, 1016.617306, 54, 180.220391, 78 169 045,
    50.64, 680.92448, 1016.581286, 55, 180.216065, 78 762 217, 39.76, 687.612458, 1020.953228, 56,
    180.218351, 78 533 125, 47.21, 689.092235, 1030.568799, 57, 180.222303, 80 061 160, 36.91, 702.320221,
    1023.731539, 58, 180.216374, 76 963 780, 48.98, 673.342631, 1024.408693, 59, 180.222377, 77 706 584,
    49.24, 680.739508, 1032.03989, 60, 180.224753, 80 341 133, 37.64, 705.723709, 1025.237636};
ss65536 = {1, 180.427853, 79 005 293, 39.34, 689.723142, 1009.636095, 2, 180.212971, 78 832 220, 36.15,
    688.611311, 1010.131339, 3, 180.206597, 78 103 460, 49.36, 680.193591, 1021.199022, 4, 180.216417,
    78 121 624, 43.7, 679.990616, 1024.278408, 5, 180.231439, 77 167 862, 53.22, 673.770564, 1024.734528,
    6, 180.49983, 78 776 428, 42.5, 687.222522, 1012.185171, 7, 180.205821, 78 586 579, 45.76, 681.578379,
    1017.565031, 8, 180.218922, 78 449 820, 38.92, 686.605611, 1024.296866, 9, 180.229719, 80 529 427,
    37.92, 704.209942, 1004.342943, 10, 180.237526, 79 589 549, 36.08, 701.036421, 1015.728154, 11,
    180.20742, 78 224 955, 43.21, 684.528931, 1026.163308, 12, 180.213234, 78 596 561, 46.18, 688.299357,
    1023.824064, 13, 180.216509, 77 644 240, 46.86, 681.075452, 1025.663765, 14, 180.230916, 77 391 869,
    52.48, 672.928691, 1014.228898, 15, 180.22039, 80 008 769, 36.66, 701.596337, 1009.341726, 16,
    180.225595, 79 244 330, 39.9, 693.728531, 1015.95401, 17, 180.263715, 78 849 666, 45.01, 689.104238,
    1022.533255, 18, 180.228062, 78 368 709, 45.83, 685.476784, 1022.530918, 19, 180.222991, 77 583 793,
    47.56, 677.830953, 1021.4053, 20, 180.224289, 78 375 419, 45.51, 684.996858, 1018.202858, 21,
    180.269582, 80 544 103, 37.16, 702.639177, 1008.738086, 22, 180.265778, 78 578 095, 46.24, 688.937254,
    1024.797086, 23, 180.225335, 79 764 398, 38.02, 698.592794, 1006.95512, 24, 180.219189, 79 585 724,
    40.87, 694.139474, 1011.089739, 25, 180.367732, 78 991 494, 42.39, 695.967191, 1020.425961, 26,
    180.207966, 78 573 644, 42.52, 689.830121, 1022.1463, 27, 180.228964, 78 078 124, 46.81, 682.521237,
    1023.359067, 28, 180.223206, 78 982 025, 47.4, 691.393885, 1017.857047, 29, 180.222929, 78 999 639,
    39.81, 694.515411, 1020.368048, 30, 180.225921, 78 090 112, 45.98, 683.440093, 1019.215653, 31,
    180.218674, 78 427 014, 39.66, 687.198522, 1023.570267, 32, 180.225082, 79 112 951, 34.85, 697.164008,
    1019.364381, 33, 180.217952, 78 281 323, 41.76, 684.863875, 1012.604103, 34, 180.219145, 79 163 950,
    40.18, 693.190608, 1011.759338, 35, 180.200706, 77 473 274, 44.24, 673.15366, 1017.260665, 36,
    180.206138, 79 706 179, 36.76, 696.617095, 1019.813602, 37, 180.250814, 78 662 683, 51.93, 687.534473,
    1020.184903, 38, 180.21417, 78 881 203, 41.03, 692.988648, 1018.655532, 39, 180.320094, 78 163 441,
    41.28, 685.673757, 1019.280808, 40, 180.223584, 77 344 889, 44.11, 679.386708, 1030.162143, 41,
    180.310671, 80 069 747, 42.28, 698.47181, 1017.863295, 42, 180.213795, 78 757 549, 52.72, 691.781822,
    1026.405785, 43, 180.205013, 78 471 540, 41.94, 686.865577, 1018.031258, 44, 180.220742, 78 833 878,
    39.68, 689.10323, 1010.306641, 45, 180.216823, 79 311 659, 42.61, 696.85106, 1017.165998, 46,
    180.222645, 78 139 330, 48.2, 682.019311, 1024.72662, 47, 180.300793, 79 151 748, 45.23, 694.562408,
    1027.352977, 48, 180.391127, 80 263 981, 37.68, 709.803082, 1021.996977, 49, 180.216213, 79 511 533,
    39.75, 699.489653, 1021.453255, 50, 180.26257, 80 192 087, 35.03, 704.266935, 1014.044386,
    51, 180.226415, 77 640 118, 53.53, 682.391253, 1020.719593, 52, 180.219639, 78 170 028, 38.67,
    683.45009, 1015.789978, 53, 180.205745, 80 122 821, 36.84, 701.542341, 1012.073465, 54, 180.442426,
    78 606 558, 43.91, 688.565315, 1025.320524, 55, 180.263737, 79 545 331, 42.96, 700.608483, 1017.7414,
    56, 180.158263, 79 693 746, 38.29, 698.586795, 1016.721117, 57, 180.219271, 79 741 930, 39.84,
    694.916351, 1021.183241, 58, 180.21995, 79 116 284, 47.63, 695.084328, 1029.534948, 59, 180.210881,
    78 211 075, 48.08, 685.340806, 1023.055412, 60, 180.222937, 79 151 867, 43, 694.400432, 1011.485485};
```

Data Loading

All the data sets are contained in the above section.

In[10]:=

```
ssNum = 9;
sampleNum = 60;
ss[1] = ss256; latches[1] = 256;
ss[2]  = ss512; latches[2] = 512;
ss[3] = ss1024; latches[3] = 1024;
ss[4] = ss2048; latches[4] = 2048;
ss[5] = ss4096; latches[5] = 4096;
ss[6] = ss8192; latches[6] = 8192;
ss[7] = ss16384; latches[7] = 16 384;
ss[8] = ss32768; latches[8] = 32 768;
ss[9] = ss65536; latches[9] = 65 536;

ncols = 6;
sampleCol = 1;
elapsedTCol = 2;
lioCol = 3;
waitTCol = 4;
cpuTCol = 5;
dbtimeCol = 6;

Do[
  ssLio[ssidx] = {}; ssL[ssidx] = {}; ssSt[ssidx] = {}; ssQt[ssidx] = {}; ssRt[ssidx] = {};
  theSS = ss[ssidx];
  Table[
   elapsedT = theSS[[ncols sampleidx + elapsedTCol]];
   lioTot = theSS[[ncols sampleidx + lioCol]];
   cpuSecTot = theSS[[ncols sampleidx + cpuTCol]];
   waitSecTot = theSS[[ncols sampleidx + waitTCol]];

   λ = lioTot / (1000 elapsedT);
   St = (cpuSecTot 1000) / lioTot;
   Qt = (waitSecTot 1000) / lioTot;
   Rt = St + Qt;

   AppendTo[ssLio[ssidx], lioTot];
   AppendTo[ssL[ssidx], λ];
   AppendTo[ssSt[ssidx], St];
   AppendTo[ssQt[ssidx], Qt];
   AppendTo[ssRt[ssidx], Rt];

   , {sampleidx, 0, sampleNum - 1}
  ];
  , {ssidx, ssNum}
 ];
Length[ssLio[6]]
Take[ssLio[6], 5]
Mean[ssL[6]]
```

Out[29]= 60

Out[30]= {74 774 687, 74 174 241, 73 783 895, 73 859 573, 74 516 434}

Out[31]= 409.834

Basic Statistics

In this section I calculate the basic statistics, such as the mean and median. My objective is to ensure the data has been c lected and entered correctly and also to compare the two datasets to see if they appear to be different.

In[32]:=

```
myData = Table[
    {
     latches[ssidx], Mean[ssL[ssidx]], Mean[ssSt[ssidx]], Mean[ssQt[ssidx]], Mean[ssRt[ssidx]],
     Length[ssLio[ssidx]], N[StandardDeviation[ssL[ssidx]]], N[StandardDeviation[ssSt[ssidx]]],
     N[StandardDeviation[ssQt[ssidx]]], N[StandardDeviation[ssRt[ssidx]]]
    }, {ssidx, 1, ssNum}
   ];
toGrid = Prepend[myData,
    {"CBC\nlatches", "Avg L\n(lio/ms)", "Avg St\n(ms/lio)", "Avg Qt\n(ms/lio)", "Avg Rt\n(ms/lio)",
     "Samples", "Stdev L\n(lio/ms)", "Stdev St\n(ms/lio)", "Stdev Qt\n(ms/lio)", "Stdev Rt\n(ms/lio)"}];
Grid[
 toGrid,
 Frame →
  All]
```

Out[34]=

CBC latches	Avg L (lio/ms)	Avg St (ms/lio)	Avg Qt (ms/lio)	Avg Rt (ms/lio)	Samples	Stdev L (lio/ms)	Stdev St (ms/lio)	Stdev Qt (ms/lio)	Stdev Rt (ms/lio)
256	138.734	0.0284258	0.0288526	0.0572784	60	0.814778	0.000184135	0.000344842	0.000487636
512	210.418	0.0187991	0.0125996	0.0313987	60	1.55396	0.000152414	0.000233856	0.000352127
1024	287.358	0.0137272	0.00546732	0.0191945	60	2.32697	0.0000828219	0.000255526	0.000286022
2048	332.64	0.0118625	0.00464851	0.016511	60	10.0387	0.000398166	0.00123333	0.00160922
4096	387.139	0.0099953	0.0013012	0.0112965	60	3.1811	0.0000335773	0.0000978125	0.000103975
8192	409.834	0.00939936	0.000958364	0.0103577	60	4.13574	0.0000297841	0.0000911158	0.000086636
16 384	427.075	0.00897593	0.000731758	0.00970769	60	4.11008	0.0000380549	0.000064413	0.000066204
32 768	438.803	0.0087585	0.000533976	0.00929247	60	5.34519	0.0000263652	0.0000677016	0.0000638878
65 536	437.329	0.00875556	0.000544505	0.00930007	60	4.52967	0.0000322069	0.0000644969	0.0000645832

In this section I calculate the key parameters for understanding the impact of chaning the number of latches and chains. T columns heading are queuing theory centric, not Oracle-ease centric.

In[35]:=

```
myData = Table[
   {
    latches[ssidx],
    Mean[ssL[ssidx]], 100 * (Mean[ssL[ssidx]] - Mean[ssL[1]]) / Mean[ssL[1]],
    Mean[ssSt[ssidx]], 100 * (Mean[ssSt[ssidx]] - Mean[ssSt[1]]) / Mean[ssSt[1]],
    Mean[ssQt[ssidx]], 100 * (Mean[ssQt[ssidx]] - Mean[ssQt[1]]) / Mean[ssQt[1]],
    Mean[ssRt[ssidx]], 100 * (Mean[ssRt[ssidx]] - Mean[ssRt[1]]) / Mean[ssRt[1]],
    Length[ssLio[ssidx]]
   }, {ssidx, 1, ssNum}
  ];
toGrid = Prepend[myData, {"CBC\nlatches", "Avg L\n(lio/ms)", "%\nChange", "Avg St\n(ms/lio)",
    "%\nChange", "Avg Qt\n(ms/lio)", "%\nChange", "Avg Rt\n(ms/lio)", "%\nChange", "Samples"}];
Grid[
 toGrid,
 Frame →
  All]
```

Out[37]=

CBC latches	Avg L (lio/ms)	% Change	Avg St (ms/lio)	% Change	Avg Qt (ms/lio)	% Change	Avg Rt (ms/lio)	% Change	Samples
256	138.734	0.	0.0284258	0.	0.0288526	0.	0.0572784	0.	60
512	210.418	51.6705	0.0187991	-33.866	0.0125996	-56.3312	0.0313987	-45.1823	60
1024	287.358	107.129	0.0137272	-51.7086	0.00546732	-81.0509	0.0191945	-66.489	60
2048	332.64	139.769	0.0118625	-58.2686	0.00464851	-83.8888	0.016511	-71.1741	60
4096	387.139	179.052	0.0099953	-64.8372	0.0013012	-95.4902	0.0112965	-80.2779	60
8192	409.834	195.411	0.00939936	-66.9337	0.000958364	-96.6784	0.0103577	-81.9169	60
16 384	427.075	207.838	0.00897593	-68.4233	0.000731758	-97.4638	0.00970769	-83.0517	60
32 768	438.803	216.292	0.0087585	-69.1882	0.000533976	-98.1493	0.00929247	-83.7767	60
65 536	437.329	215.229	0.00875556	-69.1985	0.000544505	-98.1128	0.00930007	-83.7634	60

Sample Set Normality Tests

Before we can perform a standard t-test hypothesis tests on our data, we need to ensure it is normally distributed...because th is one of the underlying assumptions and requirements for properly performing a t-test.

Statistical and vIsual normality test

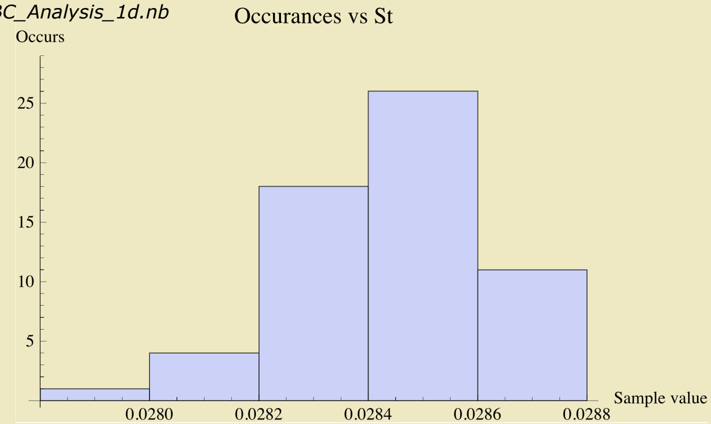
Our alpha will be 0.05, so if the distribution fit test results in a value greater than 0.05 then we can assume the data set is inde normally distributed.

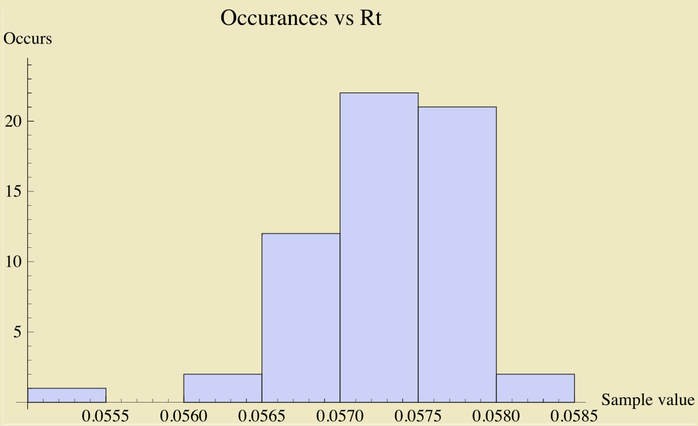
The first test is just to double check to make sure my thinking is correct. Since I creating a normal distribution based on a me and standard deviation (just happens to be based on the my sample set data), I would expect a p-value (the result) to grea exceed 0.05. Notice that the more samples I have created (the final number), the closer the p-value approaches 1.0.

In[38]:=

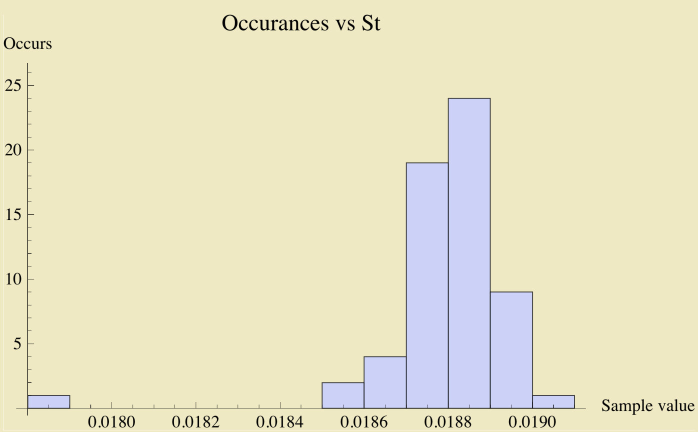
```
check = DistributionFitTest[
   RandomVariate[NormalDistribution[Mean[ssSt[1]], StandardDeviation[ssSt[1]]], 10 000]];
Do[
  pValueSt = DistributionFitTest[ssSt[i]];
  pValueRt = DistributionFitTest[ssRt[i]];
  Print["Sample set ", i, " with ",
   Length[ssSt[i]], " values. P-values: St=", pValueSt, " Rt=", pValueRt];
  st = Histogram[ssSt[i], PlotLabel → "Occurances vs St", AxesLabel → {"Sample value", "Occurs"}];
  rt = Histogram[ssRt[i], PlotLabel → "Occurances vs Rt", AxesLabel → {"Sample value", "Occurs"}];
  Print[st];
  Print[rt];
  Print["-----------------------------------------------"];
  , {i, 1, ssNum}
 ];
Print["This number should be much greater than 0.05: ", check, " If not try again by re-evaluating."];
```

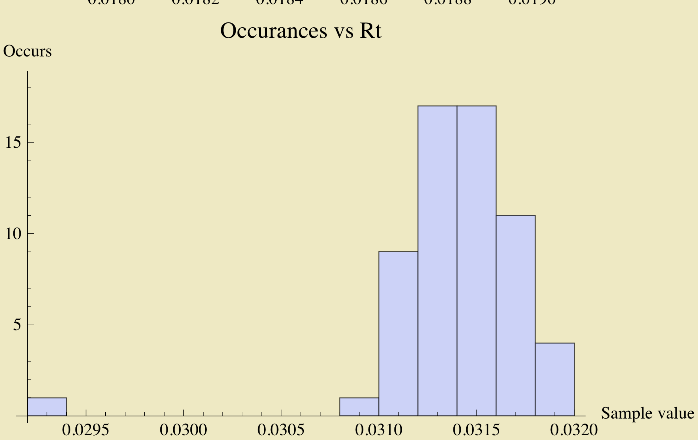
```
Sample set 1 with 60 values. P-values: St=0.0524528 Rt=0.306614
```



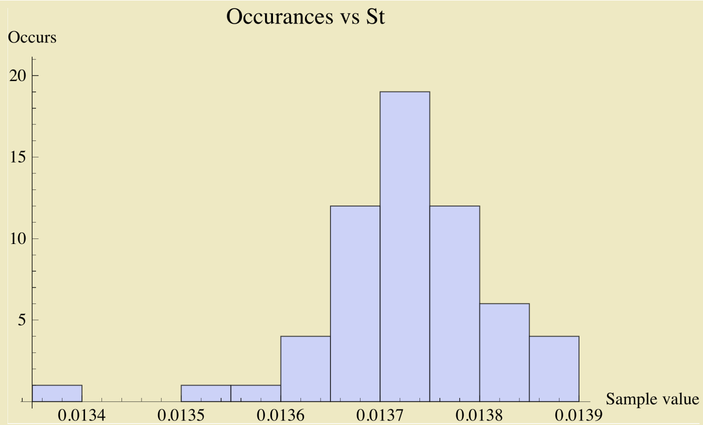


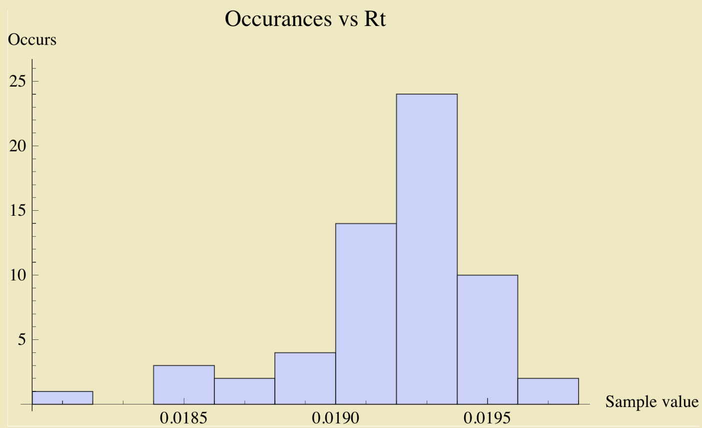

```
--------------------------------------------

Sample set 2 with 60 values. P-values: St=1.27776 × 10^-6 Rt=0.0000163863
```



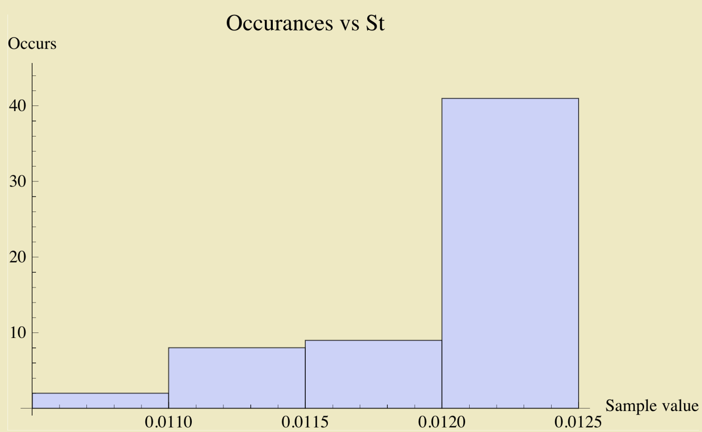


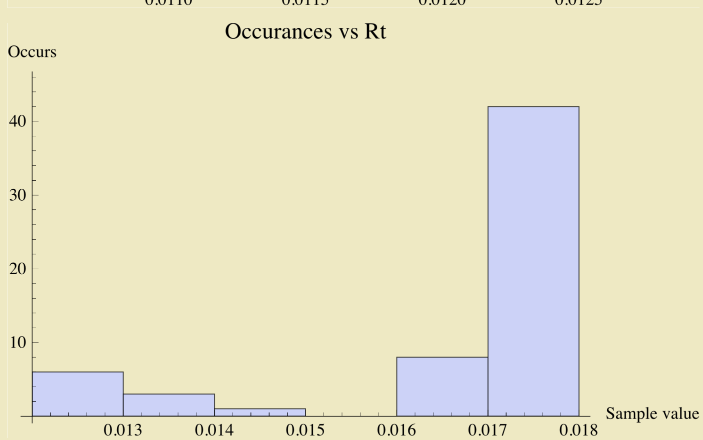

```
--------------------------------------------

Sample set 3 with 60 values. P-values: St=0.164076 Rt=0.000232992
```



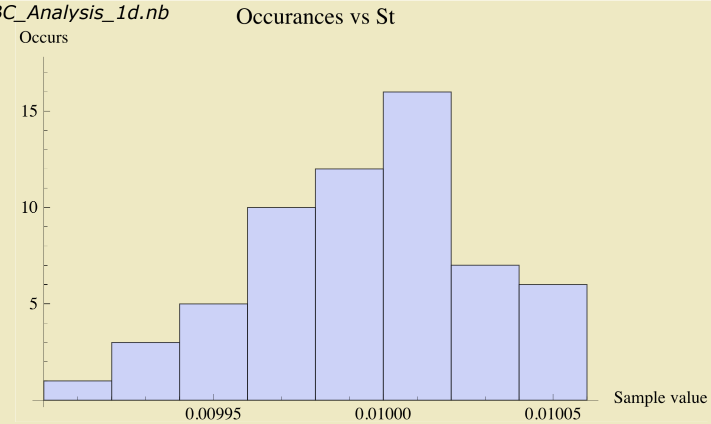


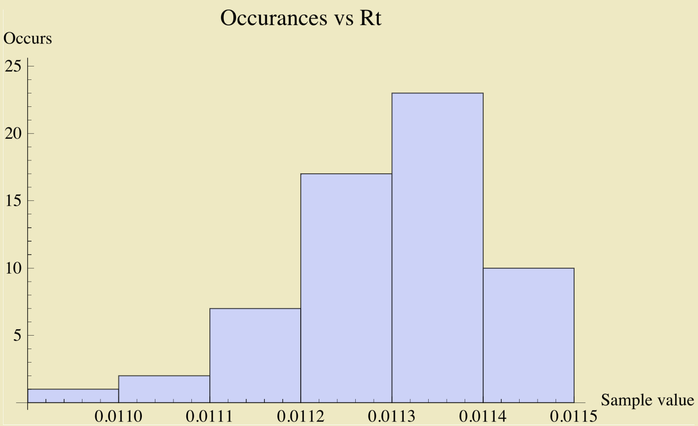

```
-------------------------------------------

Sample set 4 with 60 values. P-values: St=0. Rt=0.
```



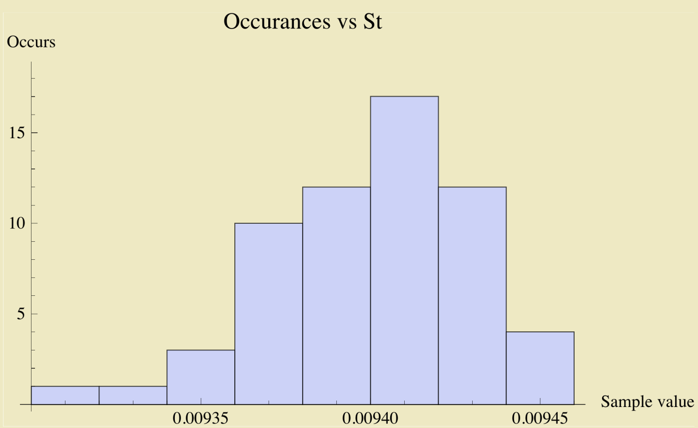


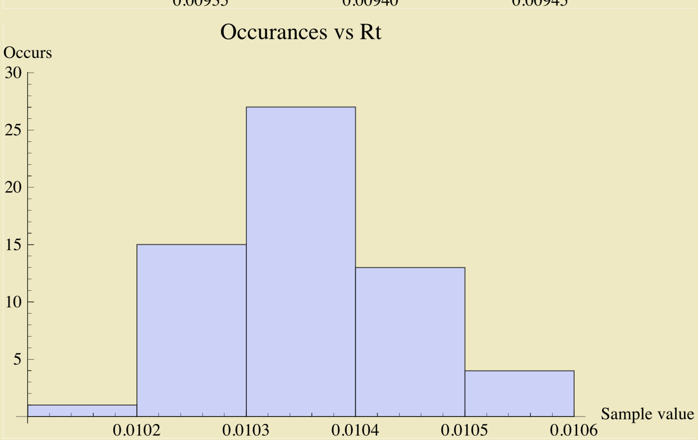

```
-------------------------------------------

Sample set 5 with 60 values. P-values: St=0.903748 Rt=0.107277
```



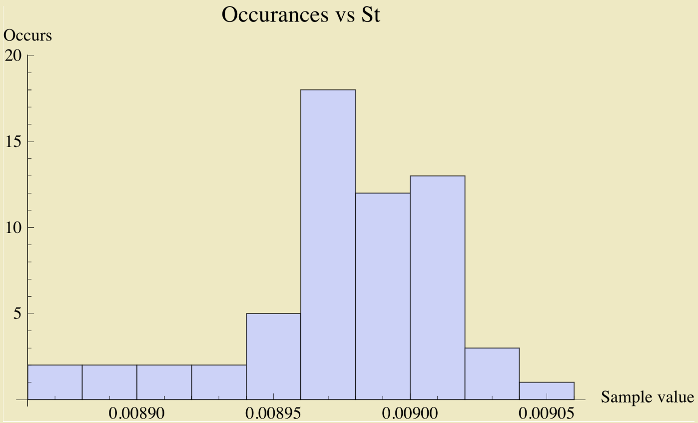


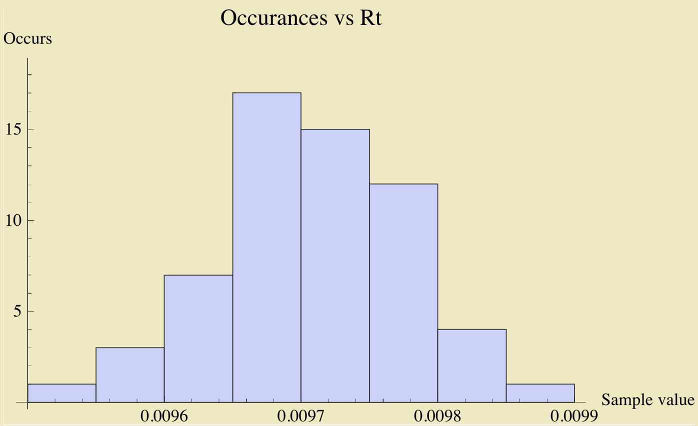

```
------------------------------------------

Sample set 6 with 60 values. P-values: St=0.189813 Rt=0.81928
```



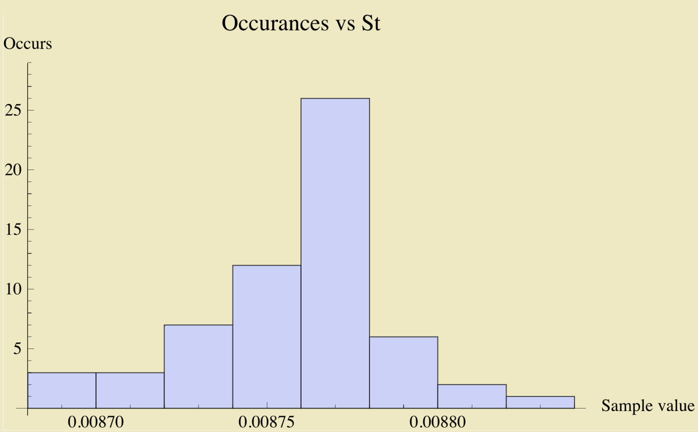


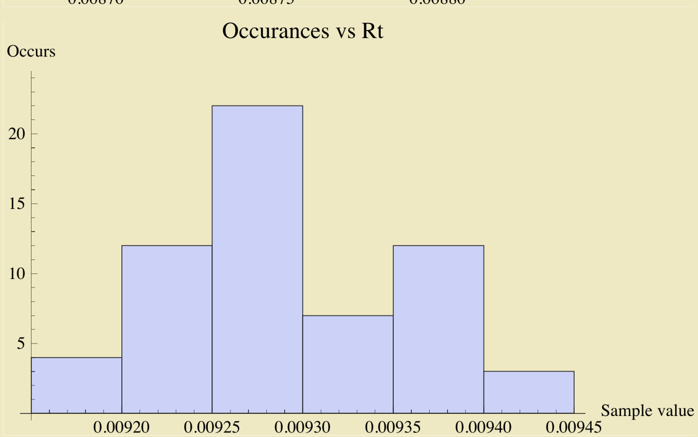

```
------------------------------------------

Sample set 7 with 60 values. P-values: St=0.00510801 Rt=0.628153
```



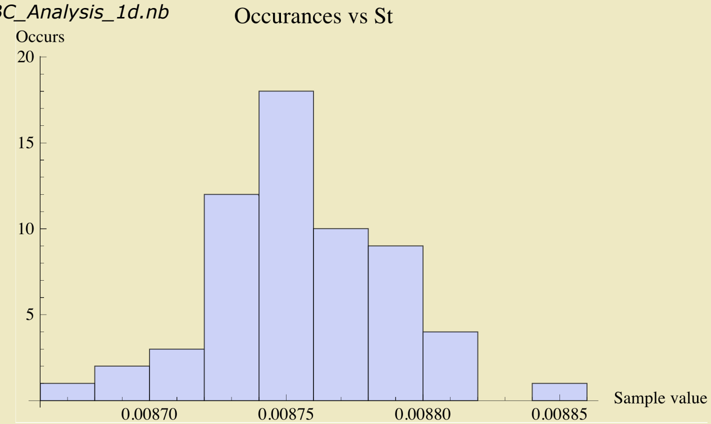


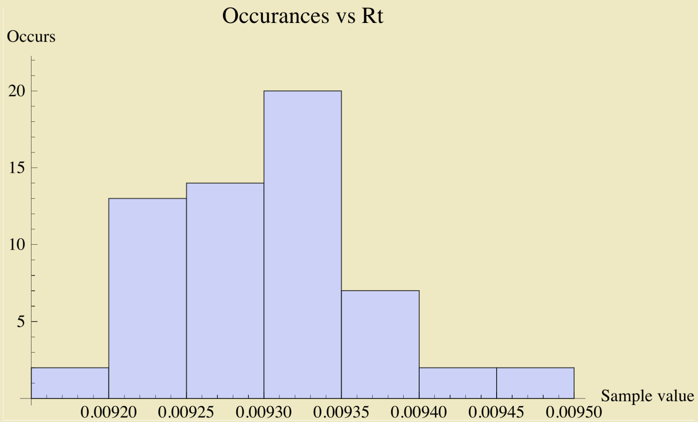

```
-------------------------------------------

Sample set 8 with 60 values. P-values: St=0.0137613 Rt=0.141916
```






```
-------------------------------------------

Sample set 9 with 60 values. P-values: St=0.897632 Rt=0.207335
```






```
-------------------------------------------

This number should be much greater than 0.05: 0.887174 If not try again by re-evaluating.
```

Sample Comparison Tests (when normality exists)

Assuming our samples **are normally distributed**, now it's time to see if they are significantly different. If so, then we kno changing the number of latches and chains indeed makes a significant performance difference...at least statistically.

The null hypothesis is; there is no real difference between our samples sets. We need to statistically prove that any difference the result of randomness; like we just happened to pick poor set of samples and it makes their difference look much worse thar really is.

A t-test will produce a statistic p. The p value is a probability, with a value ranging from zero to one. It is the answer to this que tion: If the populations really have the same mean overall, what is the probability that random sampling would lead to a differen between sample means larger than observed?

For example, if the p value is 0.03 we can say a random sampling from identical populations would lead to a difference smal than you observed in 97% of the experiments and larger than you observed in 3% of the experiments.

Said another way, suppose I have a single sample set and I copy it, resultling in two identical sample sets. Now suppose v perform a significance test on these two identical sample sets. The resuting p-value will be 1.0 because they are exactly t same. We are essentially doing the same thing here except we have to different sample sets... but we still want to see if th "like" each other..and in our case we hope they are NOT the like each other, which means the p-value will low... below our cut value of 0.05.

For our analysis we choose alpha of 0.05. To accept that our two samples are statistically similar the p value would need to less than 0.05 (our alpha).

Good reference about the P-Value and significance testing: http://www.graphpad.com/articles/pvalue.htm

Here we go (assuming our samples are normally distributed):

1. Our P value threshold is 0.05, which is our alpha.
2. The null hypothesis is the two populations have the same mean. (Remember we have to sample sets, which not the popu tion.)
3. Do the statistical test to compute the P value.
4. Compare the result P value to our threshold alpha value. If the P value is less then our threshold, we will reject the null hypoth sis and say the difference between our samples is significant. However, if the P value is greater than the threshold, we cann reject the null hypothesis and any difference between our samples are not statistically significant.

In[41]:=

```
Do[
  pValueSt = TTest[{ssSt[i], ssSt[i + 1]}];
  Print["St: (", Length[ssSt[i]],
   " values) pvalue between sample set ", i, " and ", i + 1, " is ", pValueSt];
  pValueRt = TTest[{ssRt[i], ssRt[i + 1]}];
  Print["Rt: (", Length[ssRt[i]],
   " values) pvalue between sample set ", i, " and ", i + 1, " is ", pValueRt];
  ,
  {i, 1, ssNum - 1}
 ];
```

TTest::nortst : At least one of the p-values in $\{0.0524528, 1.27776\times10^{-6}\}$, resulting from a test for normality, is below 0.025`. The tests in {T} require that the data is normally distributed. ≫

St: (60 values) pvalue between sample set 1 and 2 is 5.44313×10^{-169}

TTest::nortst : At least one of the p-values in {0.306614, 0.0000163863}, resulting from a test for normality, is below 0.025`. The tests in {T} require that the data is normally distributed. ≫

Rt: (60 values) pvalue between sample set 1 and 2 is 1.01951×10^{-163}

TTest::nortst : At least one of the p-values in $\{1.27776\times10^{-6}, 0.164076\}$, resulting from a test for normality, is below 0.025`. The tests in {T} require that the data is normally distributed. ≫

General::stop : Further output of TTest::nortst will be suppressed during this calculation. ≫

St: (60 values) pvalue between sample set 2 and 3 is 1.42298×10^{-157}

Rt: (60 values) pvalue between sample set 2 and 3 is 2.58161×10^{-153}

St: (60 values) pvalue between sample set 3 and 4 is 7.01882×10^{-44}

Rt: (60 values) pvalue between sample set 3 and 4 is 7.30389×10^{-24}

St: (60 values) pvalue between sample set 4 and 5 is 2.18046×10^{-42}

Rt: (60 values) pvalue between sample set 4 and 5 is 2.68525×10^{-33}

St: (60 values) pvalue between sample set 5 and 6 is 2.42795×10^{-117}

Rt: (60 values) pvalue between sample set 5 and 6 is 8.36543×10^{-85}

St: (60 values) pvalue between sample set 6 and 7 is 2.11358×10^{-96}

Rt: (60 values) pvalue between sample set 6 and 7 is 5.07128×10^{-74}

St: (60 values) pvalue between sample set 7 and 8 is 2.37165×10^{-61}

Rt: (60 values) pvalue between sample set 7 and 8 is 4.21973×10^{-64}

St: (60 values) pvalue between sample set 8 and 9 is 0.585614

Rt: (60 values) pvalue between sample set 8 and 9 is 0.518692

If the above T-Test results (p value) are less then our threshold we can say there is a significant difference between the t sample sets.

Sample Comparison Tests (when normality does NOT exist)

If our sample sets are **not normally distributed,** we can not perform a simple t-test. We can perform what are called locati tests. I did some research on significance testing when non-normal distributions exists. I found a very nice reference:

The paragraph below (which is from the reference above) is a key reference to what we're doing here:

...the need is evident for statistical procedures that enable us to process data of "low quality," from small samples, on variabl about which nothing is known (concerning their distribution). Specifically, nonparametric methods were developed to be used cases when the researcher knows nothing about the parameters of the variable of interest in the population (hence the nar nonparametric). In more technical terms, nonparametric methods do not rely on the estimation of parameters (such as the me or the standard deviation) describing the distribution of the variable of interest in the population. Therefore, these methods a also sometimes (and more appropriately) called parameter-free methods or distribution-free methods.

Being that I'm not a statistician but still need to determine if these sample sets are significant different, I let *Mathematica* det mine the appropriate test. Notice that one of the above mentioned tests will probably be the test *Mathematica* chooses.

Note: If we run our normally distributed data through this analysis (speically, the "LocationEquivalenceTest"), *Mathematica* shou detect this and use a more appropriate significant test, like a t-test.

Here we go with the hypothesis testing (assuming our sample sets are not normally distributed):

1. Our P value threshold is 0.05, which is our alpha.
2. The null hypotheses is the two populations have the same mean. (Remember we have to sample sets, which is not the popu tion.)
3. Do the statistical test to compute the P value.
4. Compare the result P value to our threshold alpha value. If the P value is less then our threshold, we will reject the null hypoth sis and say the difference between our samples is significant. (Which is what I'm hoping to see.) However, if the P value greater than the threshold, we cannot reject the null hypothesis and any difference between our samples are not statistica significant; randomness, picked the "wrong" samples, etc.

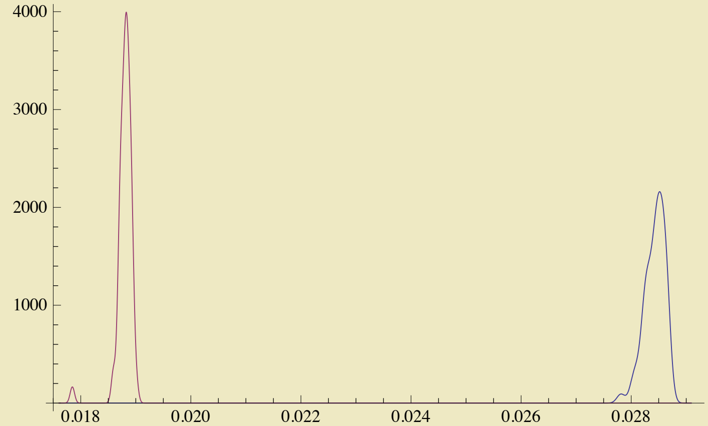
In[42]:=

```
Do[
  StHist = SmoothHistogram[{ssSt[i], ssSt[i + 1]}];
  StTest1 = MannWhitneyTest[{ssSt[i], ssSt[i + 1]}];
  StTest2 = LocationEquivalenceTest[{ssSt[i], ssSt[i + 1]}, {"TestDataTable", "AutomaticTest"}];
  Print["St: (", Length[ssSt[i]], " values) Between sample ",
   i, " and ", i + 1, ". Test1=", StTest1, " Test2=", StTest2];
  Print[StHist];
  Print["-----------------------------------------------------"];
  QtHist = SmoothHistogram[{ssQt[i], ssQt[i + 1]}];
  QtTest1 = MannWhitneyTest[{ssQt[i], ssQt[i + 1]}];
  QtTest2 = LocationEquivalenceTest[{ssQt[i], ssQt[i + 1]}, {"TestDataTable", "AutomaticTest"}];
  Print["Qt: (", Length[ssQt[i]], " values) Between sample ",
   i, " and ", i + 1, ". Test1=", QtTest1, " Test2=", QtTest2];
  Print[QtHist];
  Print["-----------------------------------------------------"];
  RtHist = SmoothHistogram[{ssRt[i], ssRt[i + 1]}];
  RtTest1 = MannWhitneyTest[{ssRt[i], ssRt[i + 1]}];
  RtTest2 = LocationEquivalenceTest[{ssRt[i], ssRt[i + 1]}, {"TestDataTable", "AutomaticTest"}];
  Print["Rt: (", Length[ssRt[i]], " values) Between sample ",
   i, " and ", i + 1, ". Test1=", RtTest1, " Test2=", RtTest2];
  Print[RtHist];
  Print[
   "------------------------------------------------------------------------------------"
  , {i, 1, ssNum - 1}
 ];
```

St: (60 values) Between sample 1 and 2. Test1= 3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

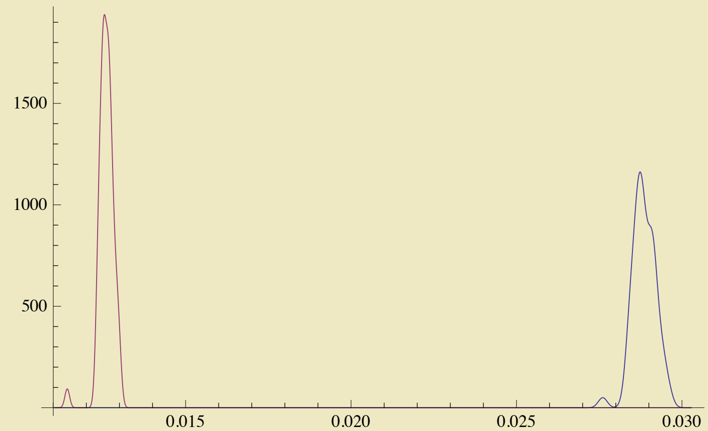
, KruskalWallis}



Qt: (60 values) Between sample 1 and 2. Test1=3.55657×10^{-21} Test2={

	Statistic	P-Value
K-Sample T	91297.5	2.54811×10^{-172}

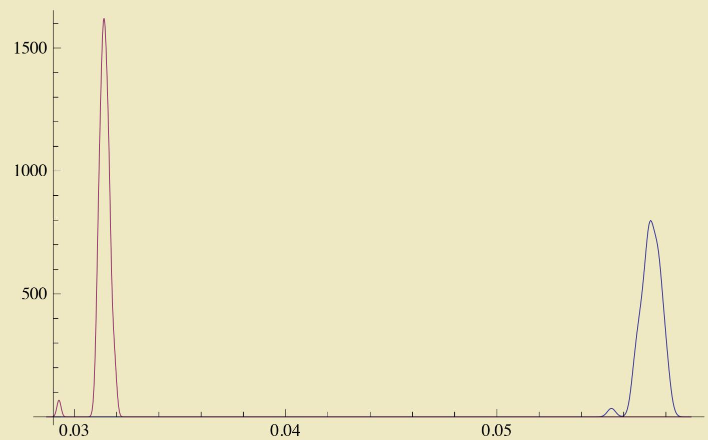
, KSampleT}



Rt: (60 values) Between sample 1 and 2. Test1=
3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

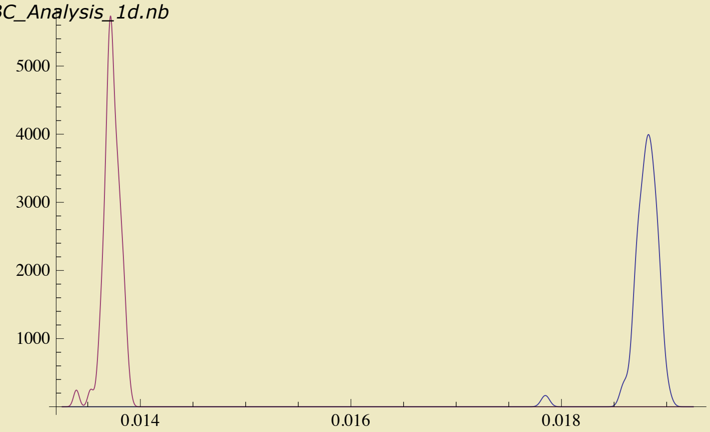
, KruskalWallis}



St: (60 values) Between sample 2 and 3. Test1=
3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

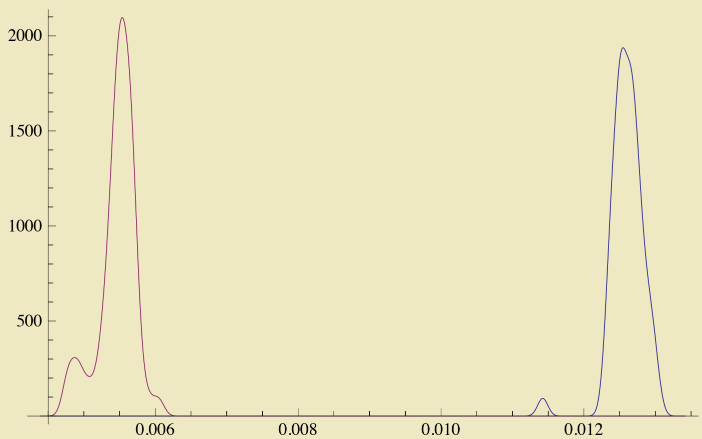
, KruskalWallis}



Qt: (60 values) Between sample 2 and 3. Test1= 3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

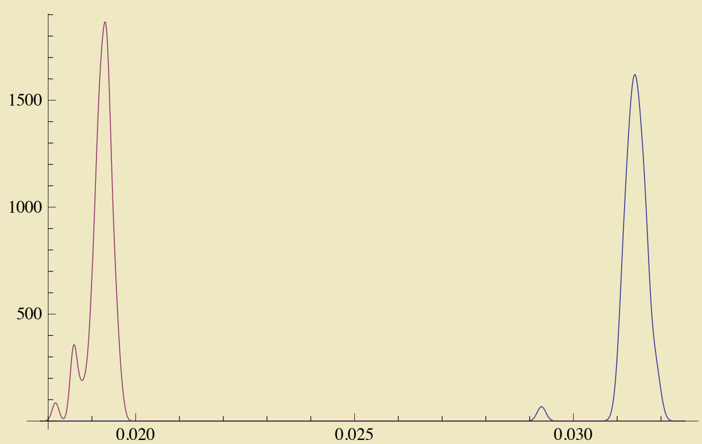
, KruskalWallis}



Rt: (60 values) Between sample 2 and 3. Test1= 3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

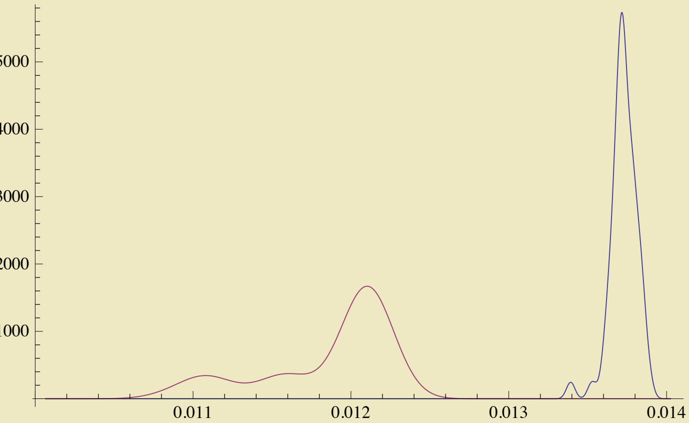
, KruskalWallis}



St: (60 values) Between sample 3 and 4. Test1= 3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

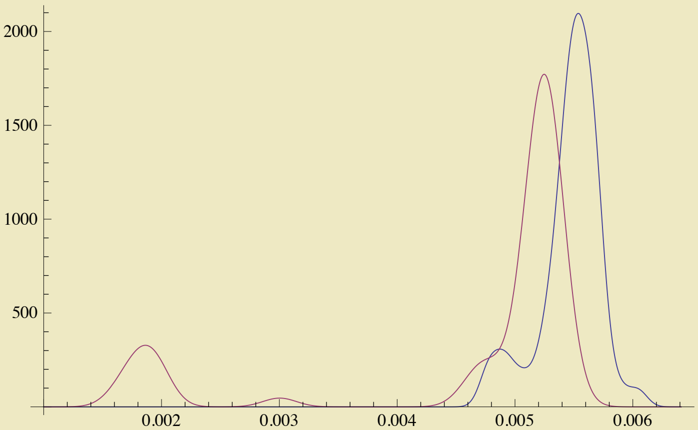
, KruskalWallis}



Qt: (60 values) Between sample 3 and 4. Test1=
2.59347×10^{-13} Test2={

	Statistic	P-Value
Kruskal-Wallis	53.5327	5.27721×10^{-17}

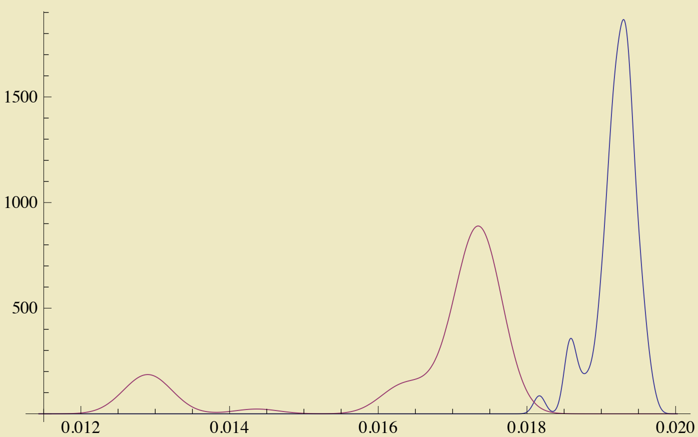
, KruskalWallis}



Rt: (60 values) Between sample 3 and 4. Test1=
3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

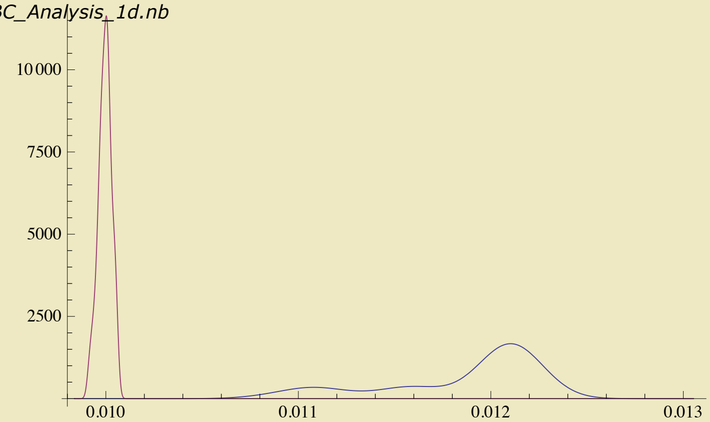
, KruskalWallis}



St: (60 values) Between sample 4 and 5. Test1=
3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

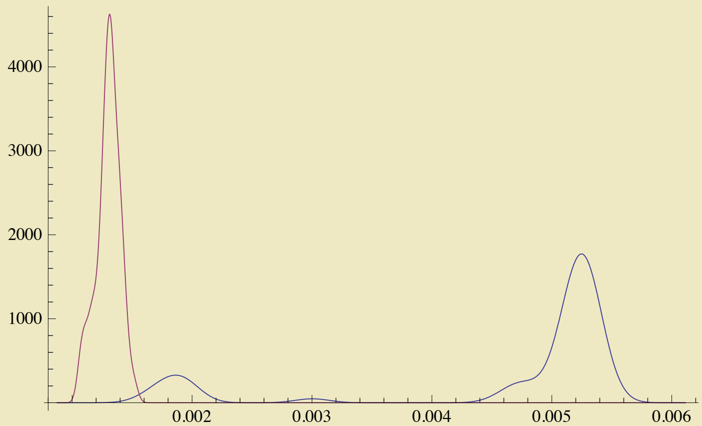
, KruskalWallis}



Qt: (60 values) Between sample 4 and 5. Test1= 3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

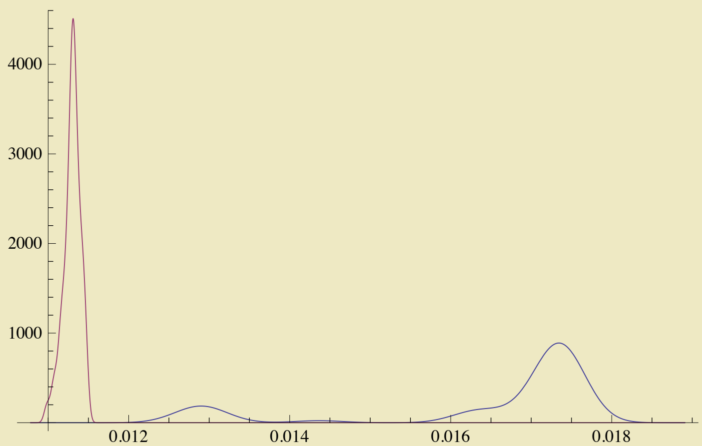
, KruskalWallis}



Rt: (60 values) Between sample 4 and 5. Test1= 3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

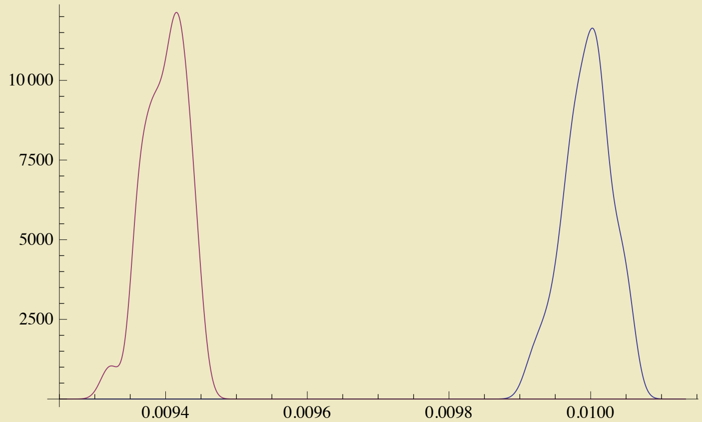
, KruskalWallis}



St: (60 values) Between sample 5 and 6. Test1=3.55657×10^{-21} Test2={

	Statistic	P-Value
K-Sample T	10577.7	2.42795×10^{-117}

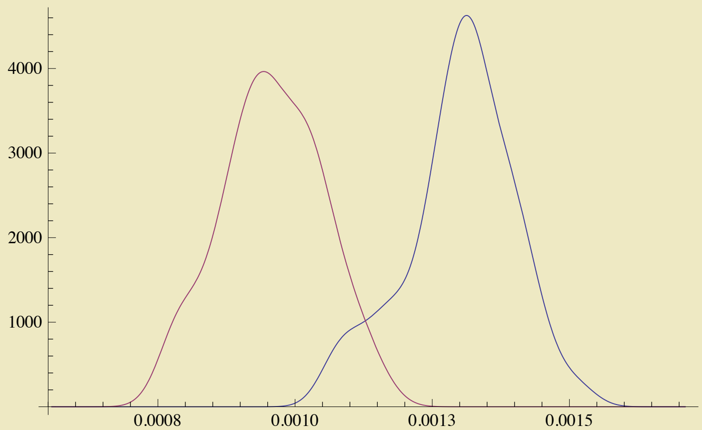
, KSampleT}



Qt: (60 values) Between sample 5 and 6. Test1=1.06416×10^{-20} Test2={

	Statistic	P-Value
K-Sample T	394.658	1.91438×10^{-39}

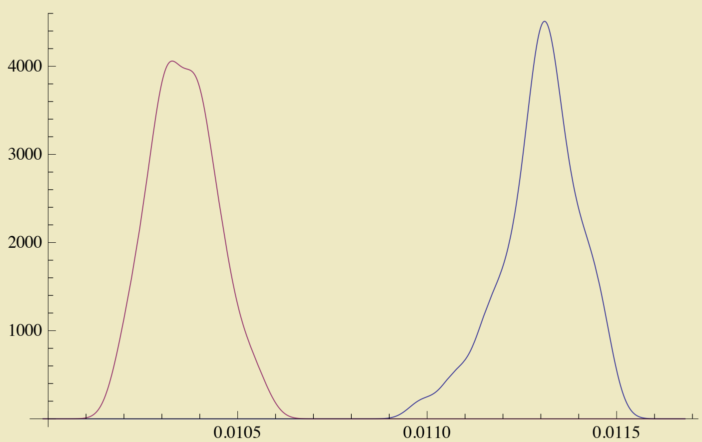
, KSampleT}



Rt: (60 values) Between sample 5 and 6. Test1=3.55657×10^{-21} Test2={

	Statistic	P-Value
K-Sample T	2886.92	8.36543×10^{-85}

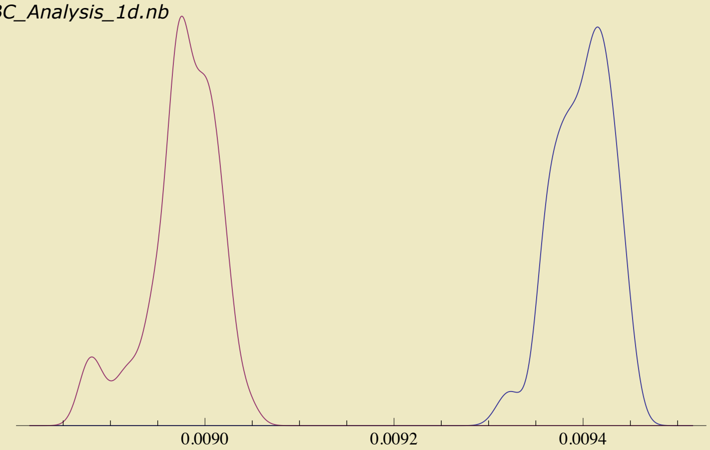
, KSampleT}



St: (60 values) Between sample 6 and 7. Test1=
3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

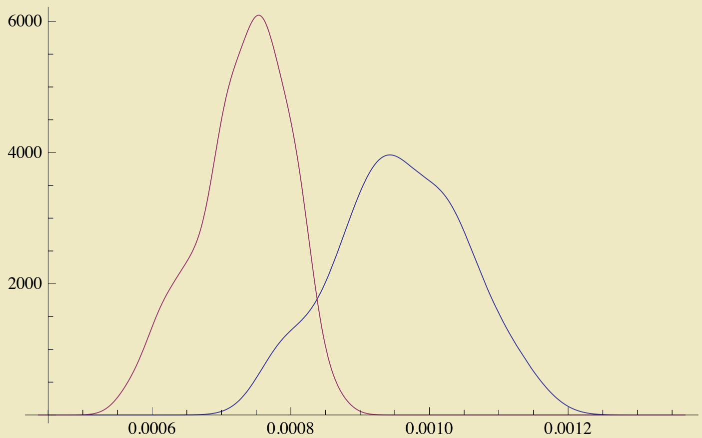
, KruskalWallis}



Qt: (60 values) Between sample 6 and 7. Test1=8.72542×10^{-20} Test2={

	Statistic	P-Value
K-Sample T	247.449	9.60099×10^{-31}

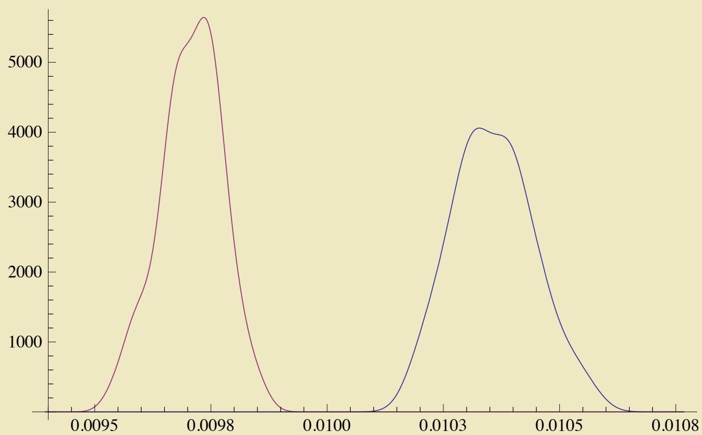
, KSampleT}



Rt: (60 values) Between sample 6 and 7. Test1=3.55657×10^{-21} Test2={

	Statistic	P-Value
K-Sample T	2132.45	2.15619×10^{-77}

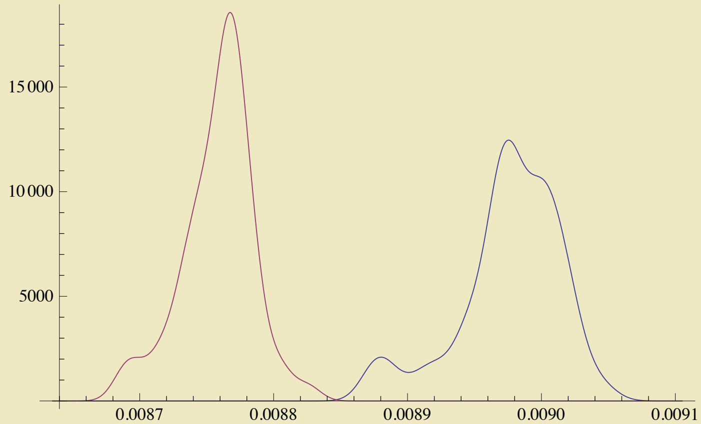
, KSampleT}



St: (60 values) Between sample 7 and 8. Test1=
3.55657×10^{-21} Test2={

	Statistic	P-Value
Kruskal-Wallis	89.2562	2.50875×10^{-37}

, KruskalWallis}

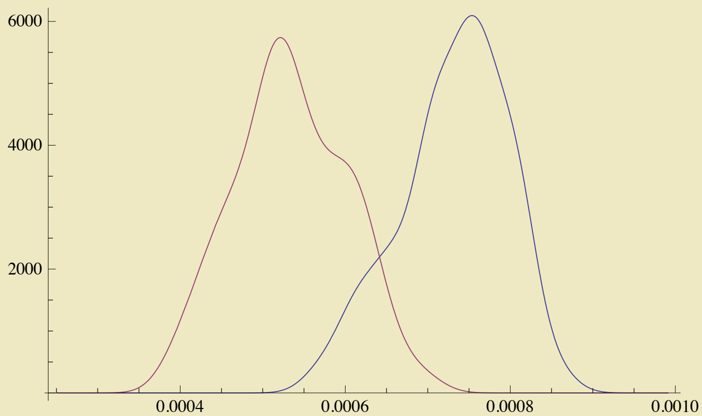



```
--------------------------------------------------------
```

`Qt: (60 values) Between sample 7 and 8. Test1=`1.28345×10^{-19} `Test2=`{

	Statistic	P-Value
K-Sample T	268.772	3.34004×10^{-32}

`, KSampleT`}

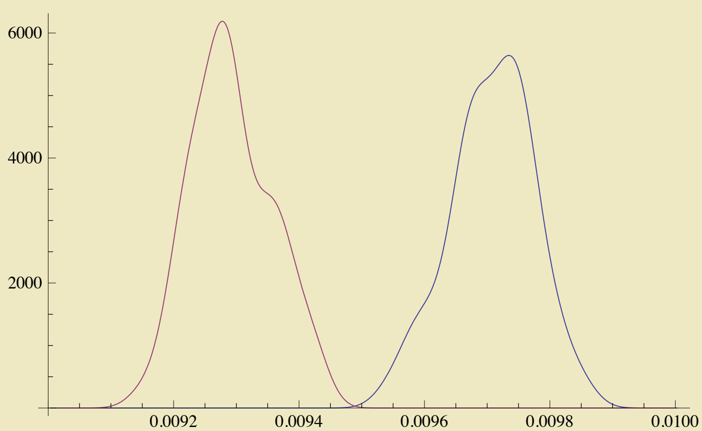



```
--------------------------------------------------------
```

`Rt: (60 values) Between sample 7 and 8. Test1=`3.55657×10^{-21} `Test2=`{

	Statistic	P-Value
K-Sample T	1222.07	4.21973×10^{-64}

`, KSampleT`}

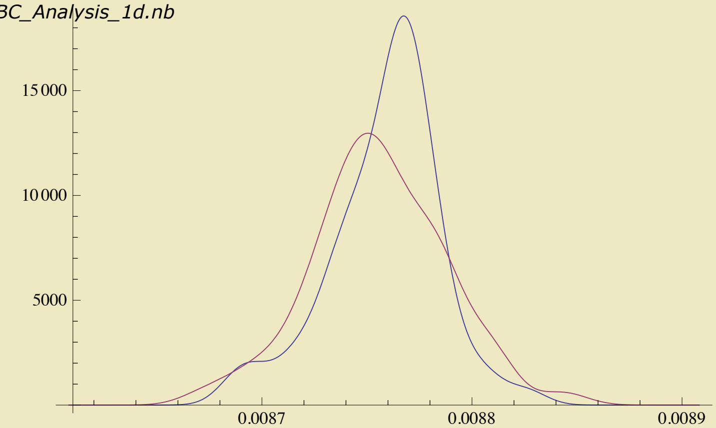



```
-----------------------------------------------------------------------------------------------------
```

`St: (60 values) Between sample 8 and 9. Test1=0.408429 Test2=`{

	Statistic	P-Value
Kruskal-Wallis	0.687713	0.409235

`, KruskalWallis`}

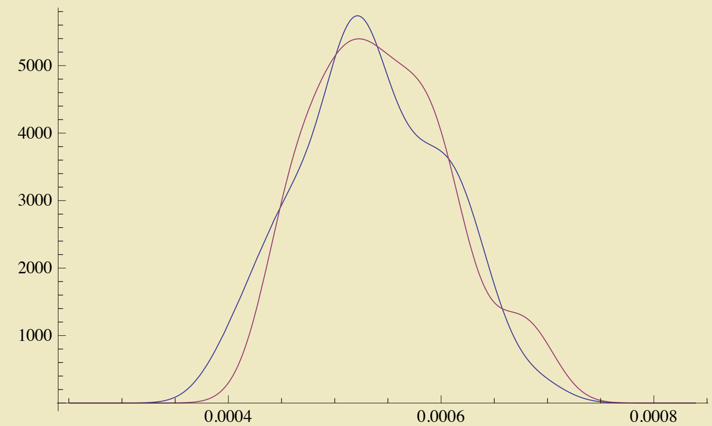



```
-------------------------------------------------------
```

```
Qt: (60 values) Between sample 8 and 9. Test1=0.496695 Test2={
```

	Statistic	P-Value
K-Sample T	0.760787	0.384854

```
, KSampleT}
```

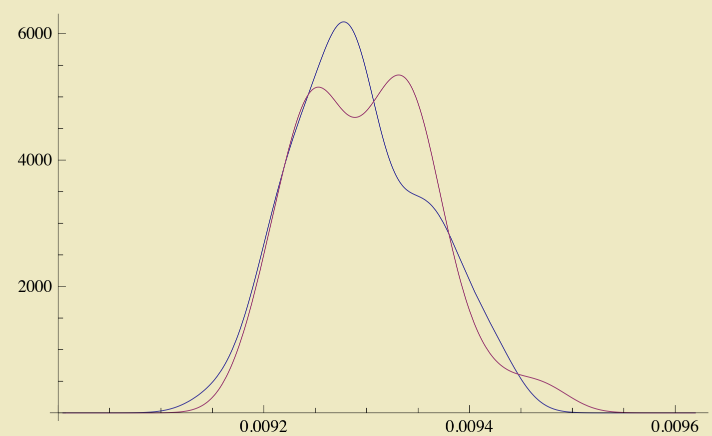



```
-------------------------------------------------------
```

```
Rt: (60 values) Between sample 8 and 9. Test1=0.64605 Test2={
```

	Statistic	P-Value
K-Sample T	0.419003	0.518692

```
, KSampleT}
```




```
---------------------------------------------------------------------------------------------------
```

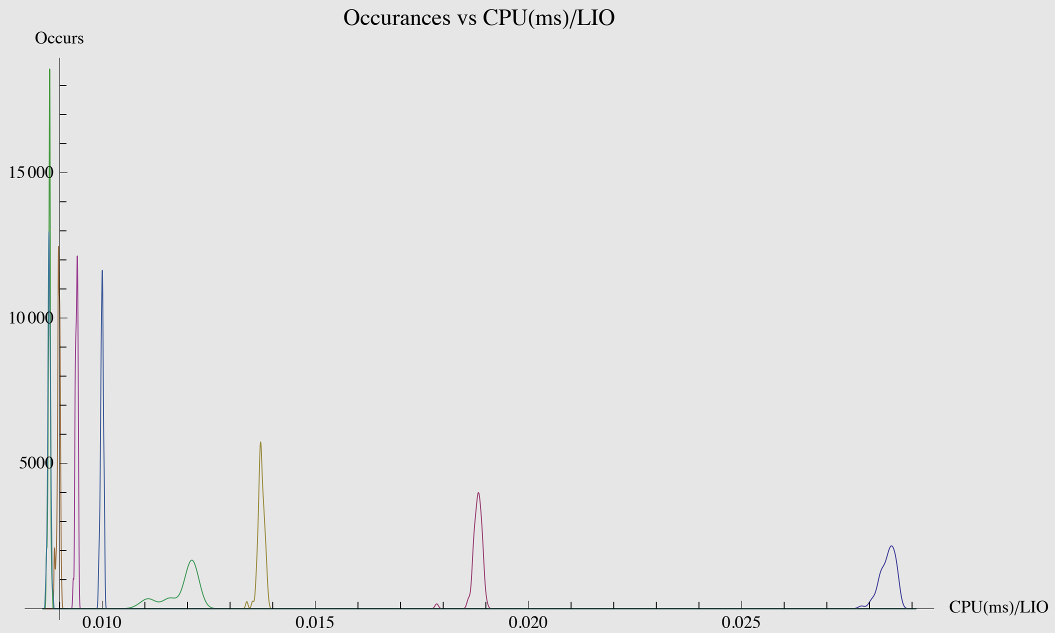
Visually Comparing All Samples

I also wanted to get a nice visual picture of my sample sets...together. Sometimes I include all the sample sets and sometime don't. It's just based on what I want to convey. Sometimes you get a more appropriate view if all the data is not included. That remove ssSt[9] for example.

In[43]:=

```
SmoothHistogram[{ssSt[1], ssSt[2], ssSt[3], ssSt[4], ssSt[5], ssSt[6], ssSt[7], ssSt[8], ssSt[9]},
 PlotLabel → "Occurances vs CPU(ms)/LIO", AxesLabel → {"CPU(ms)/LIO", "Occurs"}]
```

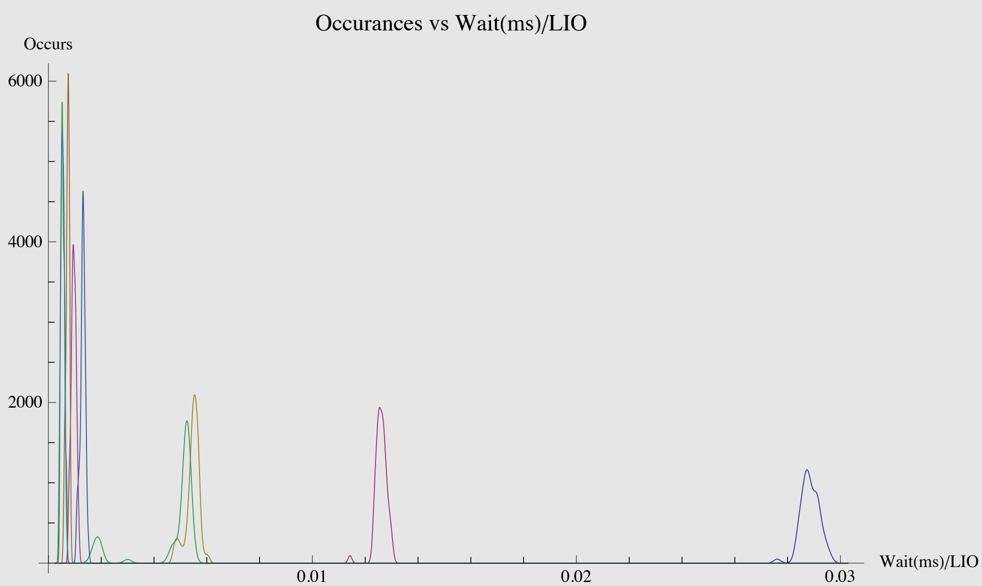
Out[43]=



In[44]:=

```
SmoothHistogram[{ssQt[1], ssQt[2], ssQt[3], ssQt[4], ssQt[5], ssQt[6], ssQt[7], ssQt[8], ssQt[9]},
 PlotLabel → "Occurances vs Wait(ms)/LIO", AxesLabel → {"Wait(ms)/LIO", "Occurs"}]
```

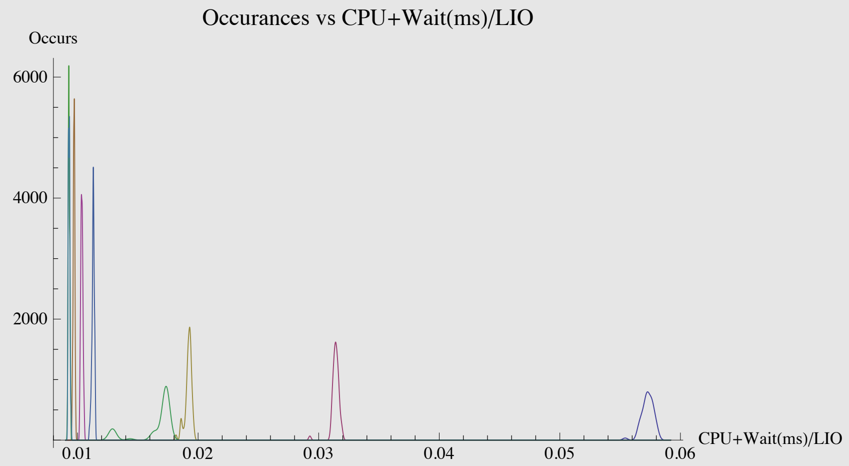
Out[44]=



In[45]:=

```
SmoothHistogram[{ssRt[1], ssRt[2], ssRt[3], ssRt[4], ssRt[5], ssRt[6], ssRt[7], ssRt[8], ssRt[9]},
 PlotLabel → "Occurances vs CPU+Wait(ms)/LIO", AxesLabel → {"CPU+Wait(ms)/LIO", "Occurs"}]
```

Out[45]=



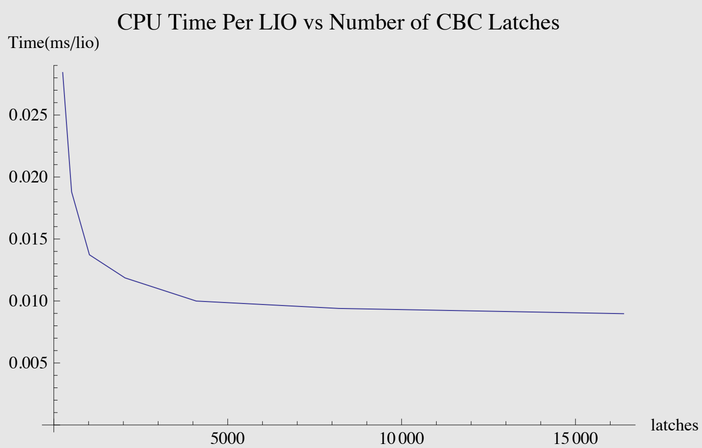
Important Graphics

I also wanted to get a nice visual picture of my sample sets...together. Sometimes I include all the sample sets and sometime don't... settting the leaveOut variable easily controls this. It's just based on what I want to convey.

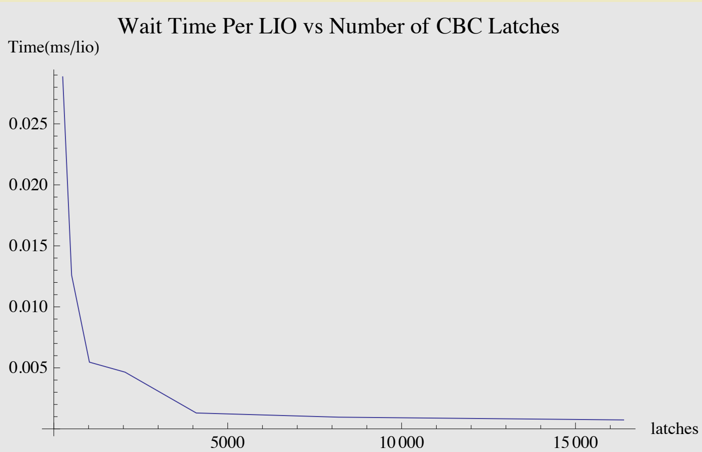
In[46]:=

```
leaveOut = 2;
St = Table[
    {latches[i], Mean[ssSt[i]]}
    , {i, ssNum - leaveOut}
  ];
Qt = Table[
    {latches[i], Mean[ssQt[i]]}
    , {i, ssNum - leaveOut}
  ];
Rt = Table[
    {latches[i], Mean[ssSt[i]] + Mean[ssQt[i]]}
    , {i, ssNum - leaveOut}
  ];
ListPlot[St, AxesOrigin → {0, 0}, PlotStyle -> PointSize[Large], Joined → True,
 AxesLabel → {"latches", "Time(ms/lio)"}, PlotLabel → "CPU Time Per LIO vs Number of CBC Latches"]
ListPlot[Qt, AxesOrigin → {0, 0}, PlotStyle -> PointSize[Large], Joined → True,
 AxesLabel → {"latches", "Time(ms/lio)"}, PlotLabel → "Wait Time Per LIO vs Number of CBC Latches"]
ListPlot[{St, Rt}, AxesOrigin → {0, 0}, PlotStyle -> PointSize[Large], Joined → True,
 AxesLabel → {"latches", "Time(ms/lio)"}, PlotLabel → "CPU+Wait Time Per LIO vs Number of CBC Latches"]
```

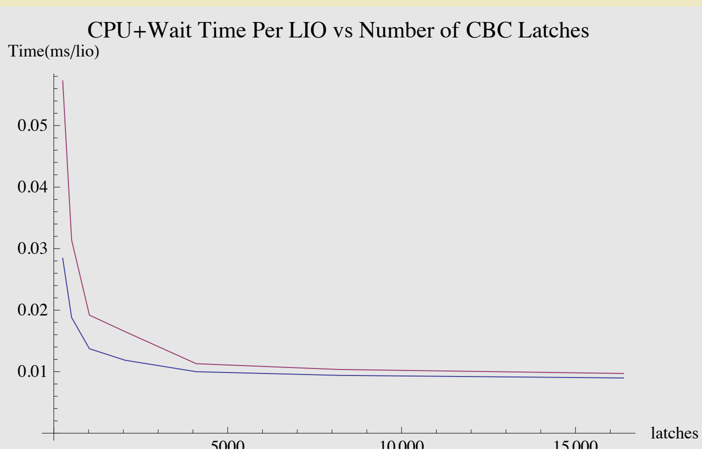
Out[50]=



Out[51]=



Out[52]=



In[53]:=

```
leaveOut = 2;
ListPlot[
 Table[
  {{latches[i], Mean[ssSt[i]]},
   {latches[i], Mean[ssSt[i]] + Mean[ssQt[i]]}
  }
  , {i, ssNum - leaveOut}
 ], AxesOrigin → {0, 0}, PlotStyle -> PointSize[Large], Joined → True, AxesLabel → {"latches", "Time(ms)"}
]
```

Out[54]=