

Reverse Conducting Evaluation

Craig Stuart Sapp <craig@ccrma.stanford.edu>

12 October 2005 -- 31 October 2005

This *Mathematica* notebook examines the individual reverse conducting performances compared to carefully corrected versions of the average taps for two performances of Chopin's Mazurka in F minor, Op. 7, No. 3 as played by (1) Ignaz Friedman in 1930 which is available on CD from Philips in the Great Pianists of the Twentieth Century, vol. 30, and (2) Charles Rosen in 1989 which is available on CD from Globe Records.

The Friedman recording is representative of a historical recording and the Rosen recording of a modern recording. The Friedman recording was reverse conducted first, and also it was manually corrected first, so there could be significant leaning effects when comparing the two recordings, although it should be noted that the Friedman performance is faster and somewhat more difficult to follow beat by beat.

The manually corrected values of the two recordings will be used to evaluate the automatically identified beat timings in terms of historic and modern recordings. For example, early recordings contain noise which may confuse automatic identification algorithms. More recent performances are also less idiosyncratic and more truer to the score (in general), so that may also effect the accuracy of an automatic identification of the beats.

Modern Recording Sample

■ Loading the raw data

First load the absolute beat positions for the average beat positions and the corrected beat positions:

```
Get["pid52932-05.corrected", Path → "D:\mazurkas\eval"]
mcbeats = pid52932x05corrected;

Get["pid52932-05.average", Path → "D:\mazurkas\eval"]
mabeats = pid52932x05average;
```

Now measure the time difference between each average and corrected beat. A negative value means that the average beat occurs before the corrected beat time.

```
mdiffs = Transpose[mabeats][[2]] - Transpose[mcbeats][[2]]

{204, 24, 0, -93, -93, -15, 38, -21, -43, 0, -18, -25, 26, -19, 0, -59, -19, 26,
-89, 15, -23, -53, -11, 19, 6, -13, 8, 81, 78, 61, 45, 10, -119, 12, 19, 51, -22,
-41, -20, -21, -65, 5, 47, -5, 0, 10, 12, 105, 46, 64, -140, -10, -8, -175, 92,
49, -296, 69, 156, 74, 115, 36, -27, 115, 103, 51, 53, 0, 11, 29, -49, -115, -23,
-32, -89, 16, 0, 0, -43, 21, -85, -16, -63, 32, -148, -95, -40, 0, 41, 10, 15,
-16, -19, -29, 0, -113, -59, -41, -130, 20, -6, 14, -93, -45, -102, -21, -25, 0,
-14, 58, 0, 41, 109, 0, 45, -106, -14, 0, 45, 113, 49, -37, 0, 41, -34, 0, 23, 0,
0, 17, -15, 16, 12, 15, 23, 35, 48, 0, -88, 0, -39, -12, -51, 49, 79, 0, -9, -9,
0, 0, 0, 38, -12, 20, 32, 23, 35, 30, 48, 76, -135, 0, 64, 28, -49, 0, -7, -97, 7,
-137, -140, 18, 33, -31, 81, 28, 37, -63, -84, -156, -33, -49, 37, 28, 7, 5, 0,
0, -9, -95, -22, -79, -88, -124, 31, 33, 30, 0, 26, 42, 20, -47, -37, -126, -25,
-124, 38, 26, 45, -32, -37, 0, -7, -88, -15, -162, -139, -32, 20, 17, -110, 0,
-107, -96, 24, -26, -103, -71, 17, -14, 0, 68, 32, -116, -43, -22, 0, -14, -34,
-88, 0, -38, 59, 123, -43, 11, -22, -40, -60, -126, -12, 27, -15, 23, 54, 26, 56,
71, 56, 0, -13, 0, 64, 95, 68, -34, 3, 83, 46, 68, 19, -29, 16, -5, -177, 0, 9,
-96, -88, 135, 78, 64, -144, 72, -11, 134, 85, 0, 34, 49, 62, 48, 107, 0, -22, 73,
-105, 15, 67, 60, 51, 44, -76, 131, 90, 70, -106, -9, -217, 19, 105, 67, -169, 88}
```

The average correction to each beat from the average tap times of all trials was 48 milliseconds. This means that, on the average, each average tap beat position must be moved 48 milliseconds to align with the correct beat time.

```
<< Statistics`ConfidenceIntervals`
```

```
Mean[Abs[mdiffs]] // N
```

```
47.9968
```

■ Manual correction of average reverse conducting beat positions

This section displays the difference between the average reverse conducted beat positions and the values derived from manually correcting these values by ear using a soundfile editor.

This corrected beat duration data was measured by listening to the audio file and manually locating the beat positions in the soundfile. The initial positions of the average reverse conducting performance were used as a baseline, and this baseline was then adjusted so that it sounded as if all beats were occurring with the attacks of notes in the performance. There are probably a few errors in the corrected due to the tediousness of getting this data, and there are a few locations in the audio where the beat positions become vague, but overall, this data represents a highly accurate description of the pianists' actual beat-by-beat performance tempo, probably an accuracy within 10 milliseconds for all beats.

```
corrected = Transpose[mcbeats][[2]]

{707, 1194, 1612, 2116, 2634, 3045, 3425, 3817, 4236, 4663, 5078, 5510, 5885, 6249,
 6661, 7106, 7570, 7916, 8442, 8784, 9320, 9897, 10439, 11145, 12060, 12659, 13010,
 13298, 13575, 14013, 14434, 14822, 15278, 15663, 15944, 16358, 16833, 17339, 17680,
 18032, 18371, 18809, 19254, 19754, 20135, 20527, 21116, 21629, 22610, 23099, 23639,
 24176, 24488, 25146, 25574, 25994, 26702, 27070, 27298, 27825, 28222, 28666, 29081,
 29397, 29628, 30092, 30543, 31036, 31391, 31777, 32357, 33038, 33612, 34016, 34550,
 34900, 35209, 35560, 35933, 36247, 36693, 36981, 37376, 37627, 38106, 38612, 38919,
 39193, 39451, 39757, 40065, 40436, 40791, 41075, 41415, 41967, 42586, 43027, 43534,
 43851, 44165, 44475, 44877, 45229, 45653, 45941, 46278, 46577, 46913, 47256, 47606,
 47850, 48050, 48415, 48675, 49134, 49427, 49703, 49968, 50229, 50646, 51050, 51390,
 51707, 52139, 52480, 52809, 53163, 53501, 53797, 54130, 54439, 54758, 55077,
 55395, 55707, 56063, 56464, 56899, 57268, 57668, 58028, 58453, 58821, 59130,
 59496, 59847, 60191, 60577, 60975, 61320, 61643, 62015, 62319, 62628, 62966,
 63269, 63595, 63925, 64217, 64882, 65368, 65753, 66151, 66599, 67012, 67501,
 68133, 68635, 69258, 69828, 70163, 70475, 70933, 71178, 71533, 71861, 72331,
 72828, 73418, 73970, 74481, 74841, 75159, 75490, 75854, 76186, 76537, 76929,
 77458, 77871, 78335, 78873, 79412, 79763, 80080, 80379, 80775, 81090, 81413,
 81776, 82249, 82702, 83270, 83776, 84351, 84712, 85057, 85364, 85777, 86142,
 86529, 86940, 87479, 87856, 88469, 89064, 89555, 90034, 90542, 91135, 91598,
 92133, 92685, 93129, 93607, 94153, 94704, 95550, 96051, 96513, 96885, 97305,
 97843, 98311, 98705, 99140, 99580, 100036, 100523, 100942, 101360, 101739, 102078,
 102625, 103022, 103557, 104001, 104496, 105131, 105753, 106577, 107665, 108289,
 108604, 108941, 109191, 109594, 110015, 110425, 110790, 111142, 111349, 111750,
 112114, 112564, 112871, 113113, 113346, 113722, 114148, 114639, 114988, 115328,
 116058, 116598, 117980, 118716, 119213, 119641, 119924, 120382, 121028, 121488,
 121967, 122279, 122543, 123080, 123543, 123957, 124322, 124662, 124896, 125466,
 125978, 126443, 126996, 127682, 127942, 128444, 128949, 129432, 130024, 130618,
 130904, 131396, 132127, 132741, 133554, 134702, 135141, 135919, 136992, 137993}
```

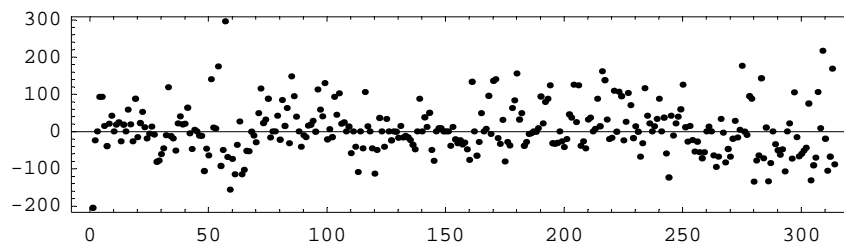
```
<< Statistics`ConfidenceIntervals`
```

```
avgtimes = Transpose[mabeats][[2]];
```

Here are the timing differences between the average reverse-conducting absolute times and the absolute times measured in a sound editor:

```
diffs = corrected - avgtimes;
```

```
ListPlot[diffs, Frame → True, AspectRatio → 1/4, PlotStyle → PointSize[0.008]];
```



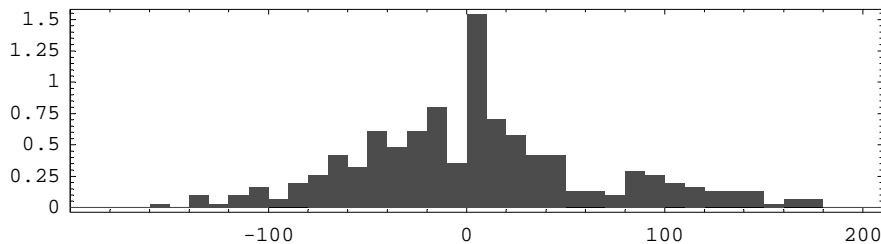
The average beats are not quite centered on the corrected beats. The average beats are, on the average, 4.6 milliseconds after the corrected beats.

```
offseterror = Mean[diffs] // N
4.55096
```

The histogram below shows the data from the plot above. It generally shows a nicely distributed range of corrections. The peak at 0 is due to the correction values being too small to change to the theoretically correct values about up to 10 milliseconds before their measured positions.

```
<< Graphics`Graphics`
```

```
Histogram[diffs, AspectRatio -> 1/4, Frame -> True, BarEdges -> False,
HistogramCategories -> 50, HistogramScale -> 100, HistogramRange -> {-200, 200}];
```



1.5% of the average reverse conducting beats were sufficiently close to the manually audible analysis of the beat positions so that they did not require alteration to align with the audio beat events. The distribution should be smooth (since about 1/2 as much is expected to be exactly on the correct beat), so there are some beats marked as 0 ms difference to the true location which are occurring slightly after the actual beat position.

■ Trial Quality Measurements

Define a logarithmic score for the amount of correction needed for a given beat. A score of 1 means 20 ms correction, 2 means 40 ms correction, 3 means 80 ms correction, 4, means 160 ms correction, 5 means 360 ms correction, *etc.* A score of 0 means there was less than 1 millisecond correction necessary, and a score of 0.5 means 10 milliseconds were needed to correct the beat location.

Qualitatively, a score of 0-1 is an "A", 1-2 is a "B", 2-3 is a "C", 3-4 is a "D", 4-5 is an "E", and 5-6 is an "F". A score of 6 means the correction was 640 milliseconds, or about 2/3 of a second.

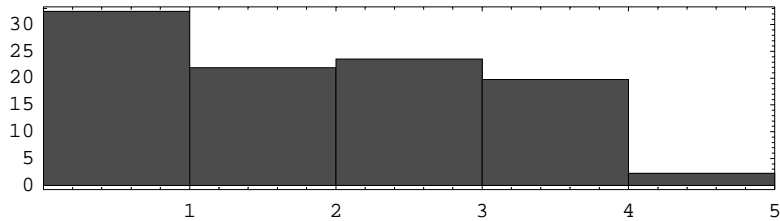
```
QScore[seq_] := Module[{sign, scores},
  sign = Map[If[# < 0, -1, If[# > 0, 1, 0]] &, seq];
  scores = Log[2.0, Abs[seq] / 10] /. -Infinity -> 0;
  scores = Map[If[# < 1, (# + 1) / 2.0, #] &, scores];
  scores = Map[If[# < 0, 0, #] &, scores];
  scores * sign
]
```

Mostly the corrections for the average of the tapping trials are evenly distributed between A, B, C, and D quality, although "A" corrections were the most common. The worst case correction was between a value of 4 and 5. This occurs on the first beat when there has been no preparation of a previous event, so is most likely due to the reaction time of the reverse conduc-

tor. Other corrections in the range between 4 and 5 are also likely due to an unprepared change in tempo which was not expected beforehand.

```
<< Graphics`Graphics`
```

```
Histogram[Abs[QScore[mdiffs]], AspectRatio -> 1 / 4,  
HistogramScale -> 100, Frame -> True, HistogramCategories -> 5];
```



The range from 0-1 can be characterized as "no audible difference" (0-20 ms); 1-2 as "slight audible difference" (20-40 ms); 2-3 "minor audible difference" (40-80 ms); 3-4 as "noticeable audible difference" (80-160 ms), 4-5 as "very noticeable audible difference" (160-320 ms), and anything higher as "extremely noticeable audible difference" (> 320 ms).

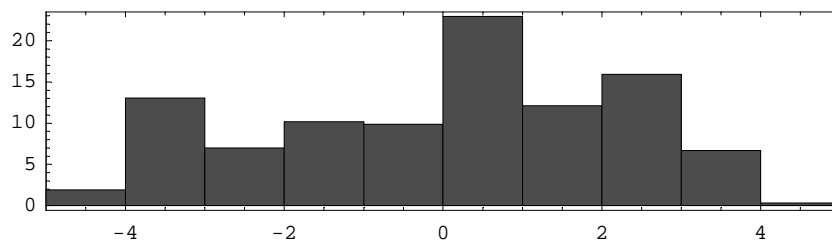
30% of the average reverse conducting beats were at the level of no audible difference, with about 20% for each of the next three categories of "slight audible difference", "minor audible difference", and "noticeable audible difference". The overall score for the average reverse conducting of the performance is a "B-":

```
Mean[Abs[QScore[mdiffs]]]
```

```
1.79979
```

Here is a histogram display of the necessary corrections with positive and negative adjustments separated. Negative values mean that the correct value occurs before the average reverse conducting time. Notice that most of the larger corrections occur in the negative range

```
Histogram[QScore[mdiffs], AspectRatio -> 1 / 4,  
HistogramScale -> 100, Frame -> True, HistogramCategories -> 11];
```



■ Examine the learning curve

How does the quality of the reverse conducting improve over time? Does the conductor learn to follow the performance better after repeated listening? First, load the individual tapping trials which contain precalculated offset values into the audio file of the performance recording:

```
Get["pid52932-05-abs.ma", Path -> "D:\mazurkas\eval"]  
data = pid52932x05abs;
```

Now calculate the timing differences between the individual trials and the corrected beat times.

```
trialdiffs = Transpose[Drop[Transpose[data], -1] - Transpose[mcbeats][[2]]];
```

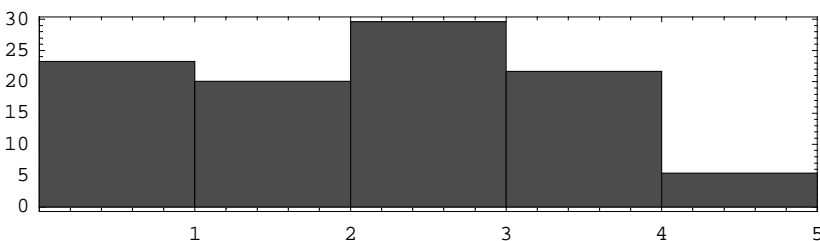
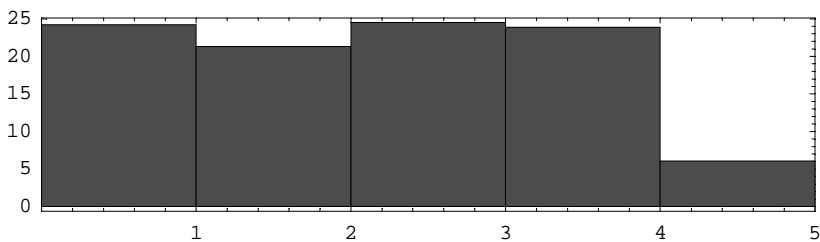
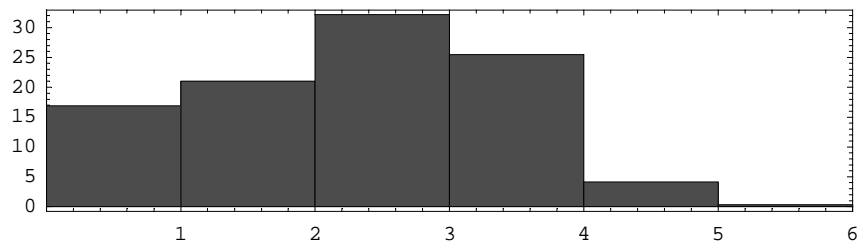
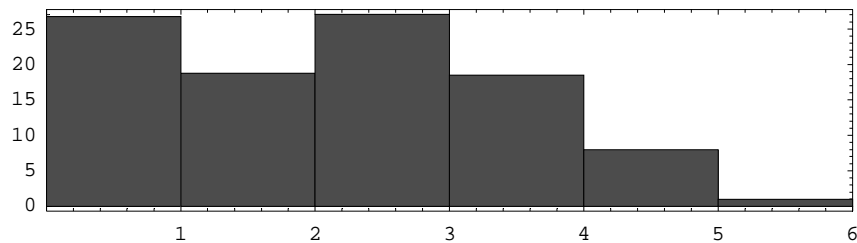
The first tapping trial gets a score of "C+", the worst trial gets a "C", and the best trial gets a "B-". The last trial is also the best trial.

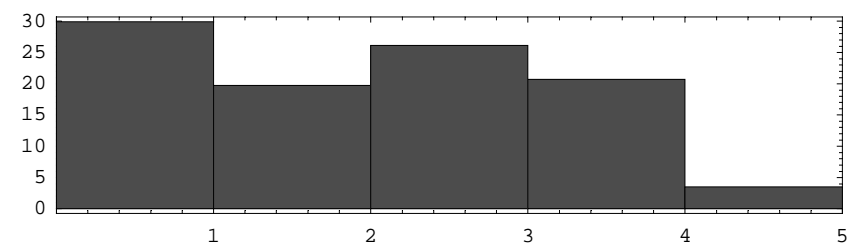
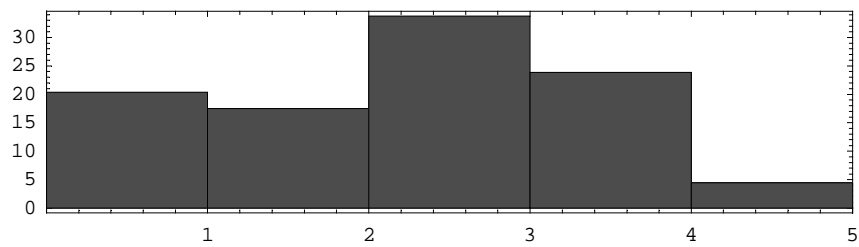
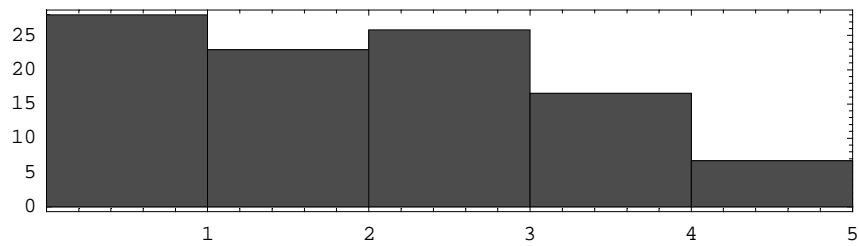
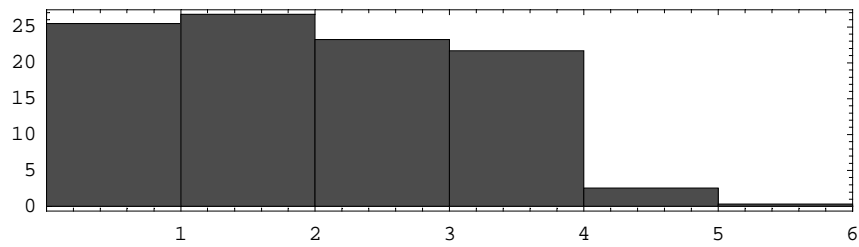
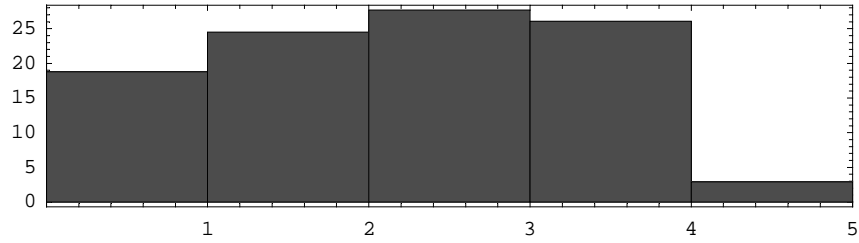
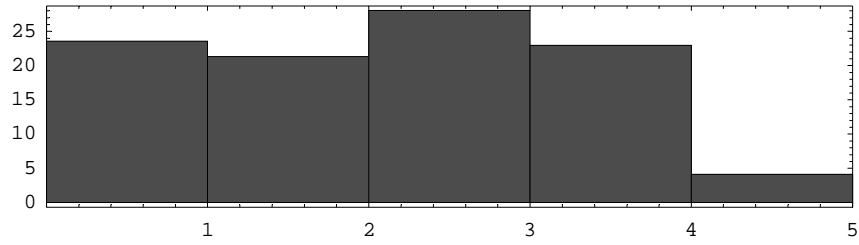
```
Map[Mean[Abs[QScore[#]]] &, trialdiffs]
```

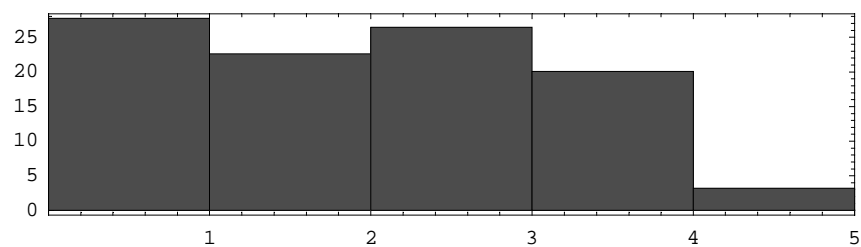
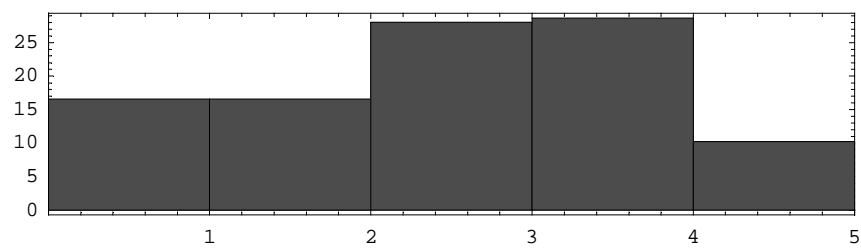
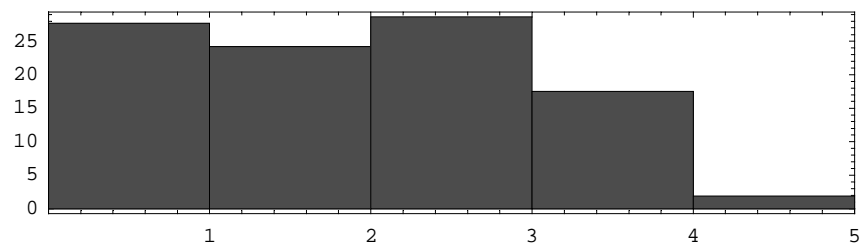
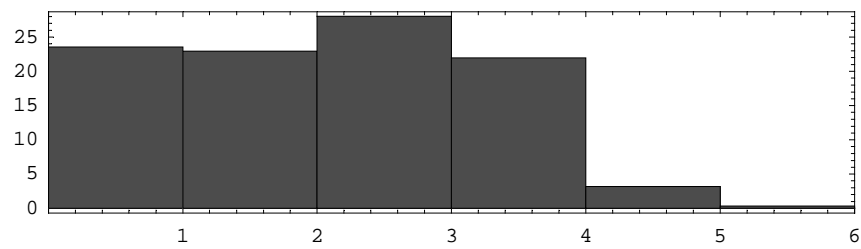
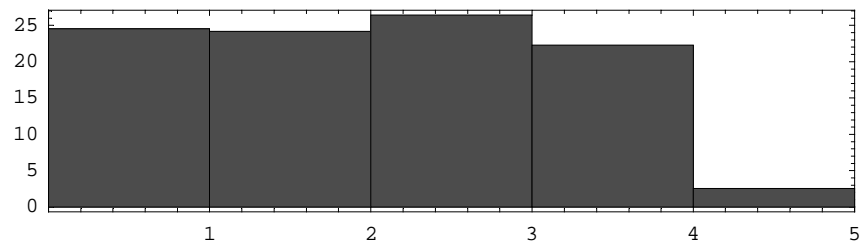
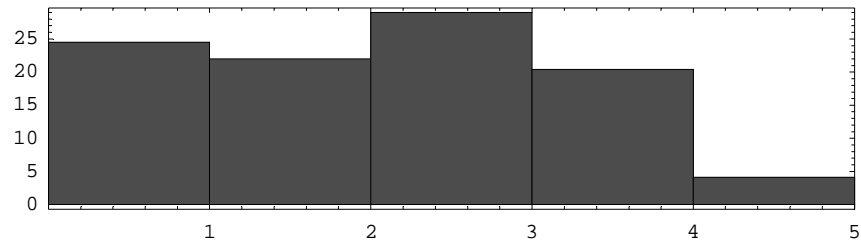
```
{2.09937, 2.29274, 2.13672, 2.12778, 2.09238, 2.19857,
 1.97603, 1.97984, 2.2208, 1.97381, 2.05081, 2.02193, 2.05213,
 1.88954, 2.45834, 1.93894, 1.93908, 2.03462, 1.94622, 1.80846}
```

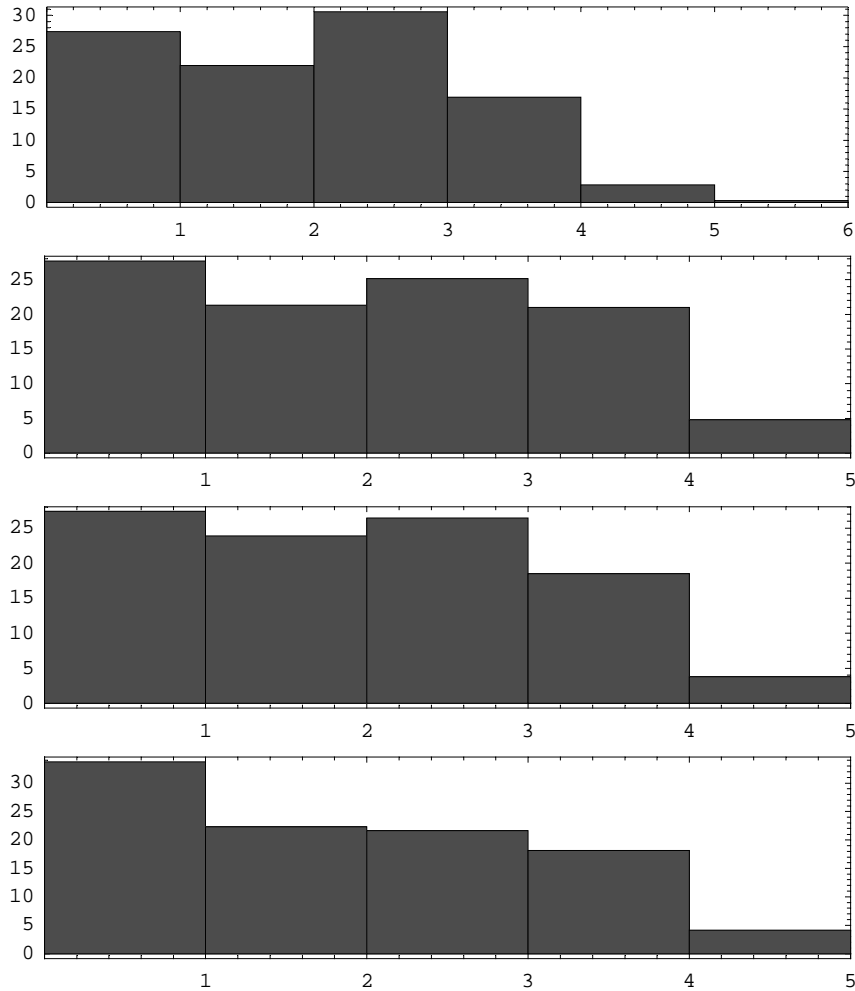
Here are quality score plots for each trial. Starting at trial 7 the categories are seen to improve so that lower categories contain a higher percentage of beats on the average.

```
Table[Histogram[Abs[QScore[trialdiffs[[xz]]]], AspectRatio -> 1/4,
  Frame -> True, Range -> {{0, 6} {0, 30}}, HistogramScale -> 100,
  HistogramCategories -> 5], {xz, 1, Length[trialdiffs]}];
```









How does the accuracy of the first half of the trials compare to the second half of the trials?:

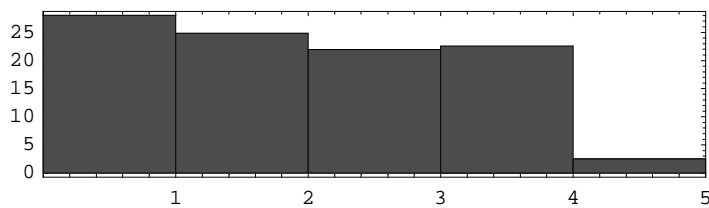
```

firsttrials = Drop[trialdiffs, -10];
lasttrials = Drop[trialdiffs, 10];

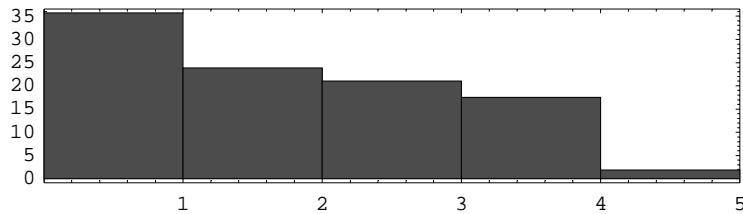
avgfirst = Map[Mean, Transpose[firsttrials]];
avglast = Map[Mean, Transpose[lasttrials]];

Histogram[Abs[QScore[avgfirst]], AspectRatio -> 1/4,
  HistogramScale -> 100, Frame -> True, HistogramCategories -> 5];

```



```
Histogram[Abs[QScore[avglast]], AspectRatio -> 1 / 4,
HistogramScale -> 100, Frame -> True, HistogramCategories -> 5];
```



So averaging only the last 10 trials of the set of 20 gives an improvement in the average correction distance to the "true" beat. About 7% of the beats move to the "A" category, the "B" category says about the same, the "C" category declines by about 3% of the total beats, and the "D" category declines by about 4% of the total beats.

The score both halves are on the opposite ends of the "B-" range:

```
Mean[Abs[QScore[avgfirst]]]
```

```
1.94785
```

```
Mean[Abs[QScore[avglast]]]
```

```
1.77254
```

On the average, beats in the second half of the trials are 6 milliseconds closer to the "true" beat locations:

```
firsterr = Mean[Abs[avgfirst]]
```

```
53.0166
```

```
lasterr = Mean[Abs[avglast]]
```

```
46.5328
```

```
firsterr - lasterr
```

```
6.48381
```

Also, averaging the performance of the second half of the trials improves the error displacement by 1.46 milliseconds on the average:

```
allerr = N[Mean[Abs[mdiffs]]]
```

```
47.9968
```

```
allerr - lasterr
```

```
1.46401
```

■ Plotting the average displacement error for each trial

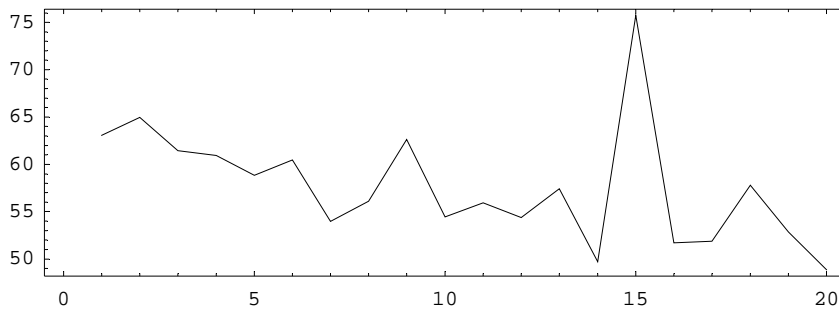
This section plots the average displacement error for each trial which shows the gradual improvement in the reverse conducting over time.

```
avgerr = Table[Apply[Plus, Sqrt[ trialdiffs[[i]] * trialdiffs[[i]] ] ] /
  Length[trialdifs[[i]]] * 1.0 , {i, 1, Length[trialdifs]}]

{63.0701, 64.9554, 61.4522, 60.9363, 58.8418, 60.4502,
  53.9926, 56.1043, 62.6264, 54.4427, 55.9139, 54.379, 57.4045,
  49.6964, 75.7357, 51.6942, 51.8568, 57.7748, 52.8652, 48.8599}
```

Note that the average correction of the averaged data was 48 milliseconds, which is better than all of the individual trials. This should be double checked, but it seems to mean that you can get better absolute positioning of the beats by averaging all trials together than by finding the best particular performance.

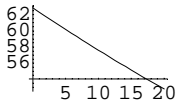
```
errplot = ListPlot[avgerr, PlotJoined → True, Frame → True, AspectRatio → 1 / 3];
```

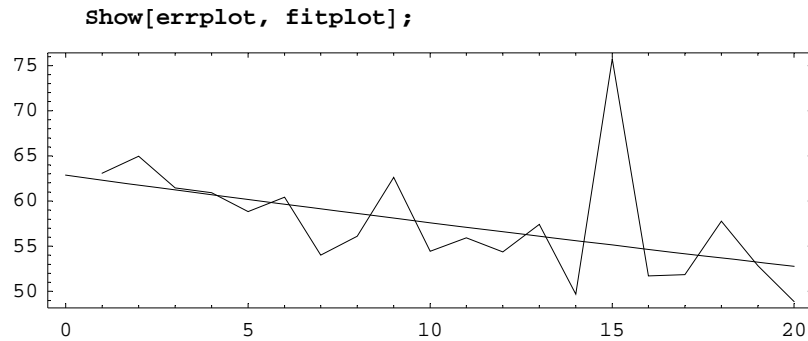


Now fit an exponentially decreasing line through the trial errors to model what the "learning curve" is for reverse conducting this performance.

```
logfit = Fit[Log[10, avgerr], {1, x}, x]
1.79842 - 0.00379873 x

fitplot = Plot[10^logfit, {x, 0, 20}];
```





Predictions of accuracy can be derived from the learning curve over time, doubling the number of trials should result in an average error of 44 ms, or 17% greater accuracy than by stopping at 20 trials.

```
err40 = 10^logfit /. x -> 40
44.3065
```

Doing 40 trials would improve the displacement error by about 2 milliseconds:

```
lasterr - err40
-44.3065 + lasterr
```

After 100 trials, the accuracy should be twice as good as after 20 trials:

```
err100 = 10^logfit /. x -> 100
26.2146
```

Doing 100 trials would improve the average displacement error by about 20 milliseconds (assuming an exponentially decaying learning curve):

```
lasterr - err100
20.3182
```

Of course, these estimations of accuracy after a certain number of trials do not take into account fatigue (see trial 15 above, for example as a possibility), and the effects of taking breaks between trials (all 20 trials were done consecutively).

■ Effect of removing the "bad" trial

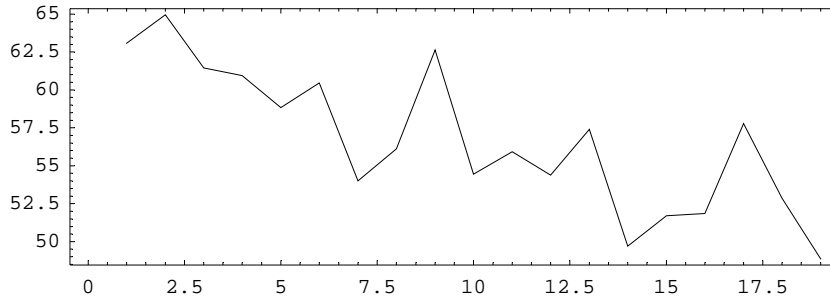
What happens if the "bad" trial (number 15 in this case) is removed from the average tapping time for each beat?

```
newerrset = RotateLeft[Drop[RotateRight[trialdiffs, 5], -1], 5];
```

```
newavgerr = Table[
  Apply[Plus, Sqrt[newerrset[[i]] * newerrset[[i]]] / Length[newerrset[[i]]] * 1.0 ,
  {i, 1, Length[newerrset]}]

{63.0701, 64.9554, 61.4522, 60.9363, 58.8418, 60.4502, 53.9926, 56.1043, 62.6264,
  54.4427, 55.9139, 54.379, 57.4045, 49.6964, 51.6942, 51.8568, 57.7748, 52.8652, 48.8599}
```

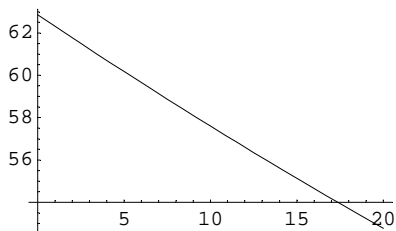
```
newplot = ListPlot[newavgerr, PlotJoined → True, Frame → True, AspectRatio → 1 / 3];
```



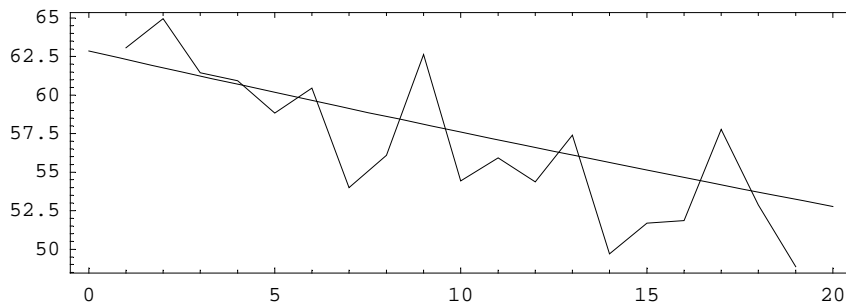
```
newlogfit = Fit[Log[10, newavgerr], {1, x}, x]
```

```
1.80374 - 0.00515696 x
```

```
newfitplot = Plot[10^logfit, {x, 0, 20}];
```



```
Show[newplot, newfitplot];
```



```
newavg = Mean[Abs[Map[Mean, Transpose[newerrset]]]]
```

```
48.3989
```

```
allerr
```

```
47.9968
```

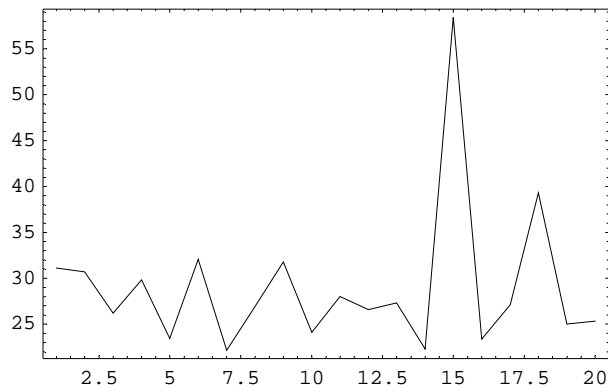
So removing the "bad" trial actually increased the overall average displacement error in this case by about 0.5 milliseconds

■ Trying to identifying "bad" performances from the average

Instead of comparing the individual performances to the corrected data, compare them to the average of all trials.

```
avgtrialdiffs = Transpose[Drop[Transpose[data], -1] - Transpose[mabeats][[2]]];
errfromavg = Map[Mean, Abs[avgtrialdiffs]] // N
{31.1148, 30.6975, 26.2006, 29.8439, 23.4501, 32.065, 22.1518, 26.898, 31.7826, 24.1242,
 28.038, 26.5892, 27.3121, 22.2919, 58.4268, 23.398, 27.119, 39.2981, 25.0075, 25.3248}

ListPlot[errfromavg, PlotJoined → True, Frame → True, PlotRange → All];
```



Trial 15 is still identified as the "worst" performance when compared to the average of all trials rather than to the corrected times. So if it is necessary to throw out "bad" performances, then this method would work in identifying the bad performances.

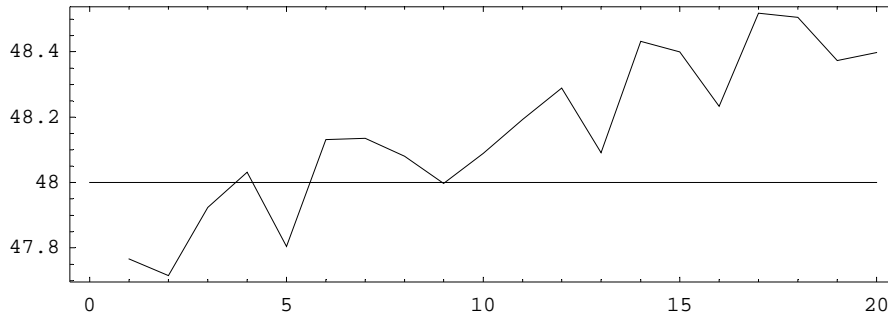
■ Dropping any single trial

Dropping the worst trial does not improve the accuracy of the average. Does dropping any other trial improve the average of all trials?

```
droppedsets = Table[RotateRight[Drop[RotateLeft[trialdiffs, i], -1], i], {i, 1, 20}];
droppedavg = Table[Mean[Abs[Map[Mean, Transpose[droppedsets][[i]]]]], {i, 1, 20}]
{47.7662, 47.7159, 47.9244, 48.0317, 47.8046, 48.1318,
 48.1355, 48.0808, 47.9971, 48.0895, 48.1934, 48.2893, 48.0915,
 48.4321, 48.3989, 48.2328, 48.5182, 48.505, 48.3727, 48.3983}

plot = ListPlot[droppedavg, PlotJoined → True,
  Frame → True, Axes → False, DisplayFunction → Identity];
```

```
Show[plot, Graphics[Line[{{0, 48}, {20, 48}}]],
      DisplayFunction -> $DisplayFunction, AspectRatio -> 1 / 3];
```



If any of trials 1, 2, 3 or 5 are dropped, then the average displacement error is reduced from the average of the rest of the trials. The horizontal line represents the displacement error when averaging all trials. So it looks like dropping earlier trials will help improve the accuracy of the average trial.

■ Dropping a range of trials

Now examine how the displacement error changes as more and more of the earlier trials are removed from the average.

```
dsets = Table[Drop[trialdiffs, i], {i, 1, 19}];

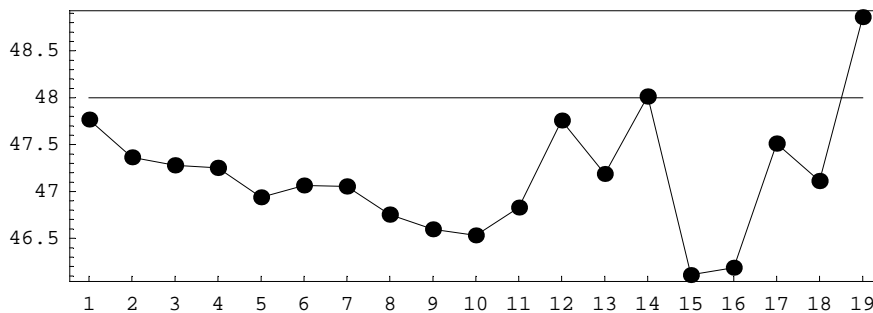
davg = Table[Mean[Abs[Map[Mean, Transpose[dsets[[i]]]]]], {i, 1, 19}]

{47.7662, 47.3665, 47.2803, 47.2526, 46.9368, 47.0649, 47.0531, 46.7516, 46.5955,
 46.5328, 46.8297, 47.7572, 47.1879, 48.0154, 46.1096, 46.189, 47.5106, 47.112, 48.8599}

dplot = ListPlot[davg, PlotJoined -> True,
  Frame -> True, Axes -> False, DisplayFunction -> Identity];

dplotp = ListPlot[davg, PlotStyle -> PointSize[0.02],
  Frame -> True, Axes -> False, DisplayFunction -> Identity];

dropstart = Show[dplot, dplotp, Graphics[Line[{{1, 48}, {19, 48}}]],
  FrameTicks -> {Range[19], Automatic, None, None},
  DisplayFunction -> $DisplayFunction, PlotRange -> All, AspectRatio -> 1 / 3];
```



Dropping the first 15 trials gives the best error of about 46 ms, which is 2 ms better than averaging all trials together. However, dropping the first 14 trials gives an equivalent error value as averaging all trials. Even dropping the first 18 trials yields a 1 ms improvement in the error rate. Dropping more than 10 trials yields more uncertainty in the result as can be seen

by the jaggedness of the plot on the right side. It seems reasonable to assume that dropping the first 10 trials due to an improved learning of the piece will lower the average displacement error of the average beat times.

In this plot, Trial 15 does seem to be a "bad" performance. It must, however, counteract other "bad" qualities of previous trials, and so must be included when earlier trials are also included.

Just for fun, here is a plot created by dropping the later trials more and more:

```
rdsets = Table[Drop[trialdiffs, -i], {i, 1, 19}];

rdavg = Table[Mean[Abs[Map[Mean, Transpose[rdsets[[i]]]]]], {i, 1, 19}]

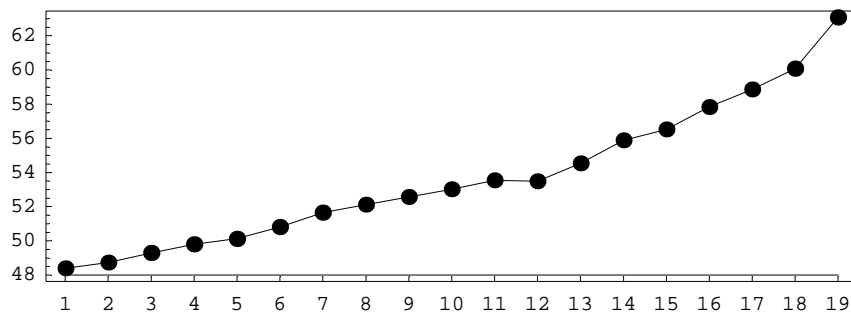
{48.3983, 48.7353, 49.3005, 49.805, 50.1293, 50.808, 51.6525, 52.1246, 52.5662, 53.0166,
 53.5471, 53.4778, 54.5349, 55.8781, 56.521, 57.8386, 58.8499, 60.0732, 63.0701}

rdplotp

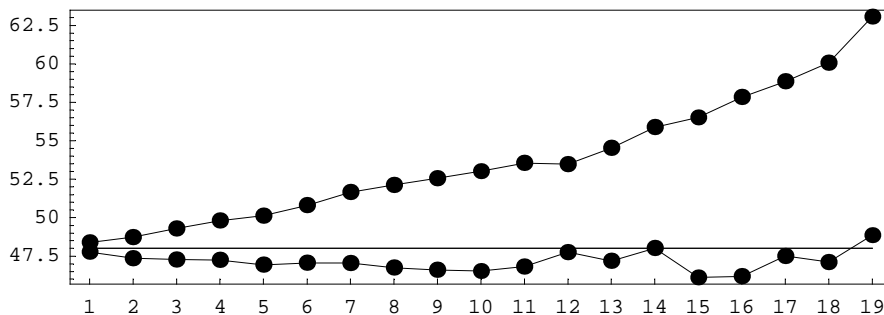
rdplotp = ListPlot[rdavg, PlotStyle -> PointSize[0.02],
  Frame -> True, Axes -> False, DisplayFunction -> Identity];

rdplot = ListPlot[rdavg, PlotJoined -> True,
  Frame -> True, Axes -> False, DisplayFunction -> Identity];

dropend = Show[rdplot, rdplotp, Graphics[Line[{{1, 48}, {19, 48}}]],
  FrameTicks -> {Range[19], Automatic, None, None},
  DisplayFunction -> $DisplayFunction, PlotRange -> All, AspectRatio -> 1 / 3];
```



```
Show[dropstart, dropend];
```



You definitely do not want to drop later trials from the average, since dropping any later trials and keeping the earlier trials will always significantly increase the average displacement error.

Also note that all individual trials are worse than trial average, as well as averages involving only the later trials:

```

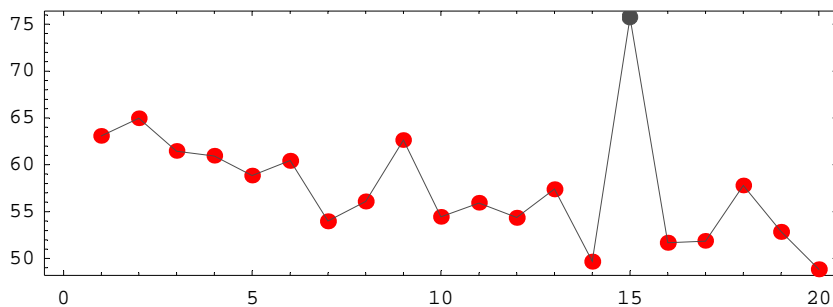
individ = Map[Mean[N[Abs[Drop[#, -1] - Transpose[mcbeats][[2]]]]] &, data]
{63.0701, 64.9554, 61.4522, 60.9363, 58.8418, 60.4502,
 53.9926, 56.1043, 62.6264, 54.4427, 55.9139, 54.379, 57.4045,
 49.6964, 75.7357, 51.6942, 51.8568, 57.7748, 52.8652, 48.8599}

individplot = ListPlot[individ, PlotStyle -> {Hue[0.0], PointSize[0.02]},
  Frame -> True, Axes -> False, DisplayFunction -> Identity, AspectRatio -> 1 / 3];

individplotp = ListPlot[individ, PlotJoined -> True, Frame -> True,
  Axes -> False, PlotStyle -> Hue[0.0], DisplayFunction -> Identity];

Show[individplot, individplotp, DisplayFunction -> $DisplayFunction];

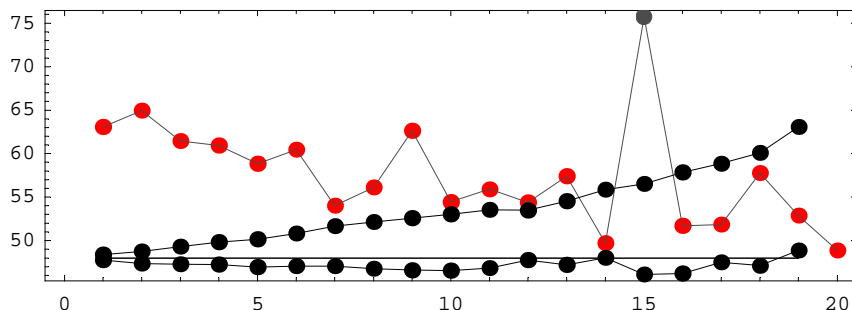
```



```

Show[individplot, individplotp,
  dropstart, dropend, DisplayFunction -> $DisplayFunction];

```



The red curve in the plot above shows the average displacement errors from the corrected times for each of the 20 individual trials. Only trials 14 and 20 are close to the error rate of the average of all trials.

■ Offset sensitivity

Individual trials were aligned to the Modern test case using a few points which seemed to be in a stable tempo region. How accurate is the offset calculated from these few points in the audio? Would the average of all trials improve if an offset is calculated for each trial based on the corrected data?

```

SquareDiff[x_, y_] := Apply[Plus, (x - y) * (x - y)]

```

```

AlignSequences[seq1_, seq2_] := Module[
  {diffs},
  diffs = Table[{x, SquareDiff[seq1, seq2 + x]}, {x, -10000, 10000, 100}];
  (x /. Solve[D[Fit[diffs, {1, x, x^2}, x], x] == 0, x])[1]
]

AlignSequences[{1, 2, 3}, {2, 3, 4}]

-1.

addoffsets = Table[
  AlignSequences[Transpose[mcbeats][[2]], Drop[data[[i]], -1]], {i, 1, Length[data]}]
{18.4695, -16.051, 1.59236, -15.7643, 1.13834, -22.4313,
 5.66382, 22.5752, -18.9027, 0.229299, -16.3107, -7.39172, 23.1146,
 6.8744, 60.0796, -2.75975, -3.64471, 36.7661, 11.909, 6.49363}

newmabeats = Drop[Map[Mean, Transpose[data + addoffsets]], -1];

extraoffset = Mean[addoffsets]

4.58249

Min[addoffsets]
Max[addoffsets]

-22.4313

60.0796

```

The average displacement error could be reduced by shifting the average trial beat times 4.6 milliseconds forward in time (total range -22 ms to +60 ms). This might be related to the 5 ms resolution of the computer keyboard. The offset of +60 may be related to poor alignment at the measured points in the audio, or perhaps even operating system multitasking. Also, this large offset error may be result of the "badness" of trial 15 (e.g., when displayed on the learning curve with other trials).

However, note that finding the individual offsets for the trials with more points, does not yield a more accurate average displacement error (48.6 compared to 48.0), so it seems that only aligning to beats in stable portions of the music is better than aligning to all beats in the music.

```

omdiffs = Transpose[mcbeats][[2]] - newmabeats;

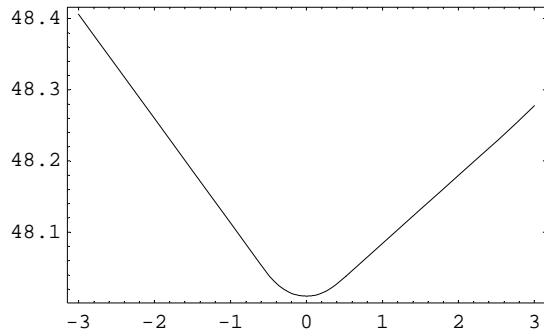
N[Mean[Abs[omdiffs]]]

48.6377

```

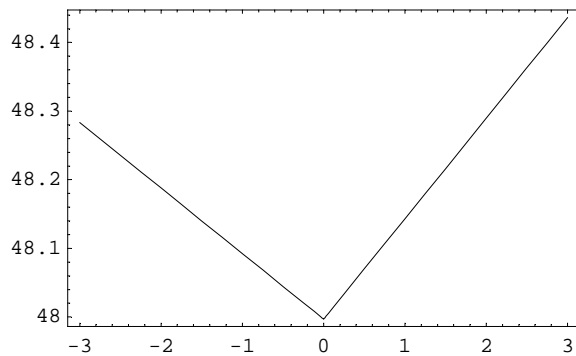
Don't know what it means, but the following plot shows that there is a wider minimum in the error rate when shifting the averaged trial differences when the offsets have been optimized for all beats in the piece, rather than a small set as shown in the second plot:

```
Plot[N[Mean[Abs[omdiffs + x + Mean[addoffsets] ]]],
     {x, -3, 3}, Frame → True, Axes → False];
```



Here is a plot using the initial offset values:

```
Plot[N[Mean[Abs[mdiffs + x ]]], {x, -3, 3}, Frame → True, Axes → False];
```



■ Comparing corrected beats to average reverse conducting beats

This section plots the average tempo range and compares it to the actual tempos measured from the audio file.

```
Get["pid52932-05.ma", Path → "D:\mazurkas\eval"]
data = Transpose[pid52932x05];

eachbeatstats = Map[DispersionReport, data] // N;

confidence =
  Map[MeanCI[#[[1]], KnownStandardDeviation → (KnownStandardDeviation /. #[[2]])] &,
      Transpose[{data, eachbeatstats}]];

durationmean = Map[Mean, data] // N;

maxdurs = Map[Max, data];

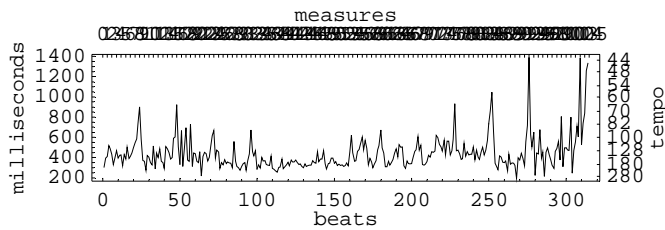
mindurs = Map[Min, data];
```

```

meanhighplot = ListPlot[Transpose[confidence][[1]], PlotJoined → True,
  DisplayFunction → Identity, PlotStyle → RGBColor[.5, .5, .5]];
meanlowplot = ListPlot[Transpose[confidence][[2]], PlotJoined → True,
  DisplayFunction → Identity, PlotStyle → RGBColor[.5, .5, .5]];
maxplot = ListPlot[maxdurs, DisplayFunction → Identity,
  PlotJoined → True, PlotStyle → RGBColor[0.8, 0.8, 0.8]];
minplot = ListPlot[mindurs, DisplayFunction → Identity,
  PlotJoined → True, PlotStyle → RGBColor[0.8, 0.8, 0.8]];

meanplot = ListPlot[durationmean,
  PlotJoined → True, PlotRange → All, AspectRatio → 1/4, Frame → True,
  FrameLabel → {"beats", "milliseconds", "measures", "tempo"},
  FrameTicks → {Automatic, Automatic, Table[{x, x/3}, {x, 0, 315, 3}],
  Map[{60000.0 / #, #} &, {30, 40, 44, 48, 54, 60, 70, 82, 100, 128, 180, 280}]}];

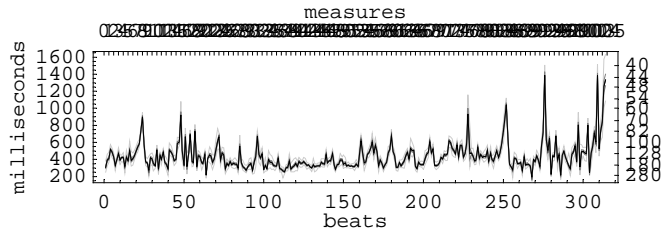
```



```

allplot = Show[meanplot, maxplot, minplot, meanhighplot, meanlowplot, meanplot];

```

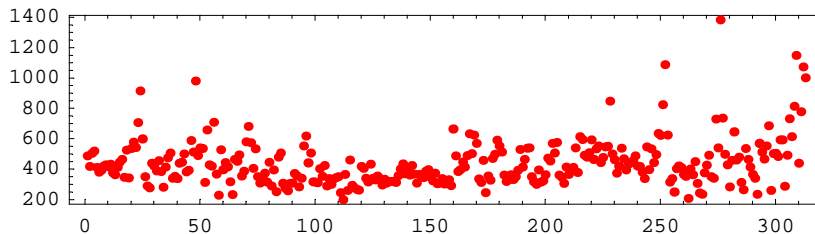


```

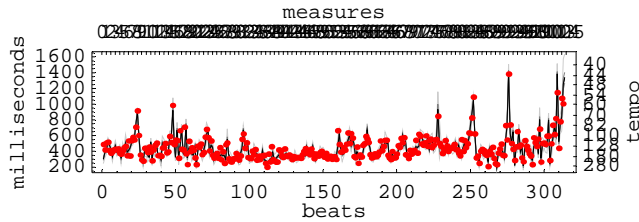
correctdeltas = Drop[RotateLeft[corrected, 1] - corrected, -1];

correctplot = ListPlot[correctdeltas, PlotStyle → {Hue[0.0], PointSize[0.012]},
  PlotRange → All, Frame → True, AspectRatio → 1/4];

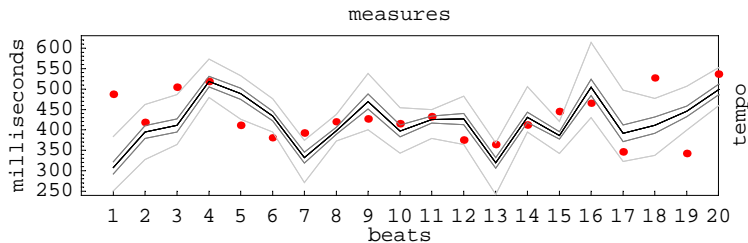
```



```
newallplot = Show[allplot, correctplot];
```



```
finalplot = Show[allplot, correctplot, allplot, Frame -> True,
  PlotRange -> {{0, 20}, {240, 630}}, FrameTicks -> {Range[20], Automatic, None, None}];
```

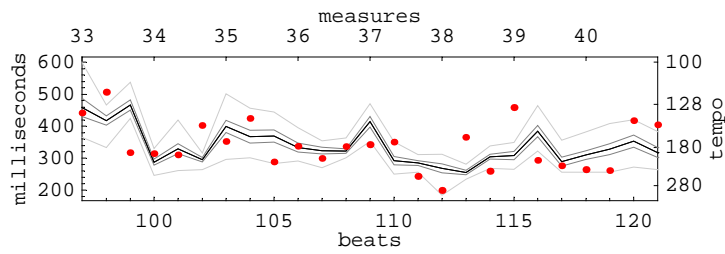
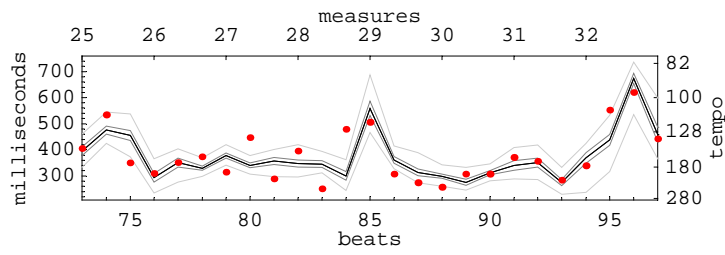
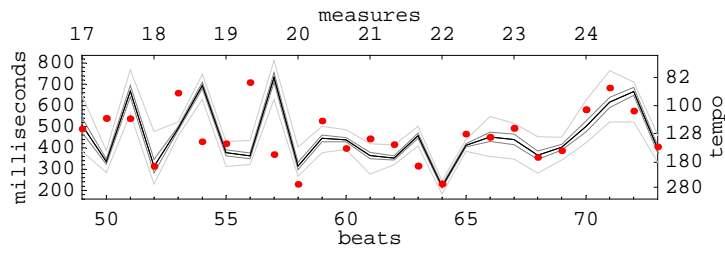
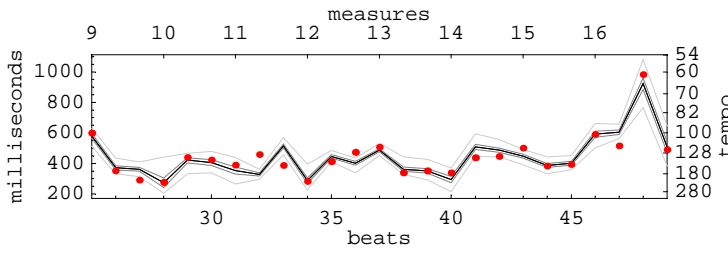
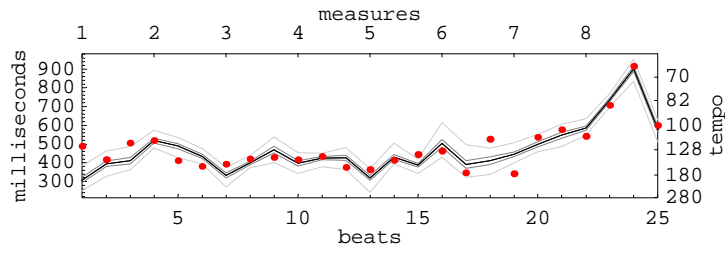


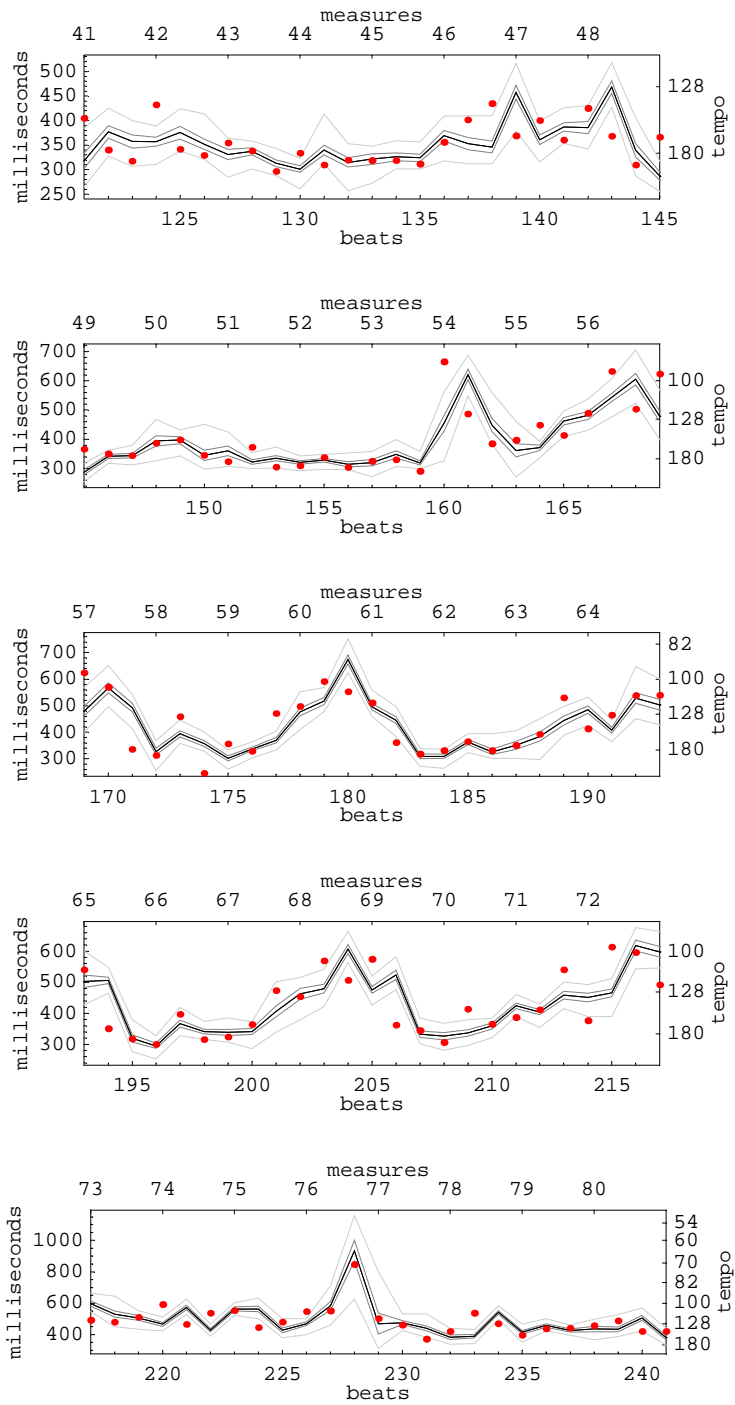
```
ShowFrame[finalplot_, minrange_, maxrange_, len_, frame_, width_] := Module[
  {framecount, startpoint, endpoint, rmin, rmax, starti, endi, delta},
  framecount = len / width + 0.99 // Floor;
  startpoint = frame * width + 1;
  endpoint = (frame + 1) * width + 1;
  starti = (startpoint - 1) + 1;
  endi = (endpoint - 1) + 1;
  If[endi > len, endi = len];
  If[starti > len, starti = len];
  rmax = Max[Take[maxrange, {starti, endi}]];
  delta = rmax * 0.03;
  rmax = rmax + delta;
  rmin = Min[Take[minrange, {starti, endi}]];
  rmin = rmin - delta;
  Show[finalplot, PlotRange -> {{startpoint, endpoint}, {rmin, rmax}},
    FrameLabel -> {"beats", "milliseconds", "measures", "tempo"},
    FrameTicks -> {Automatic, Automatic, Table[{x + 1, x / 3 + 1}, {x, 0, 315, 3}],
      Map[{60000.0 / #, #} &, {30, 40, 44, 48, 54, 60, 70, 82, 100, 128, 180, 280}]},
    Axes -> False]
]
```

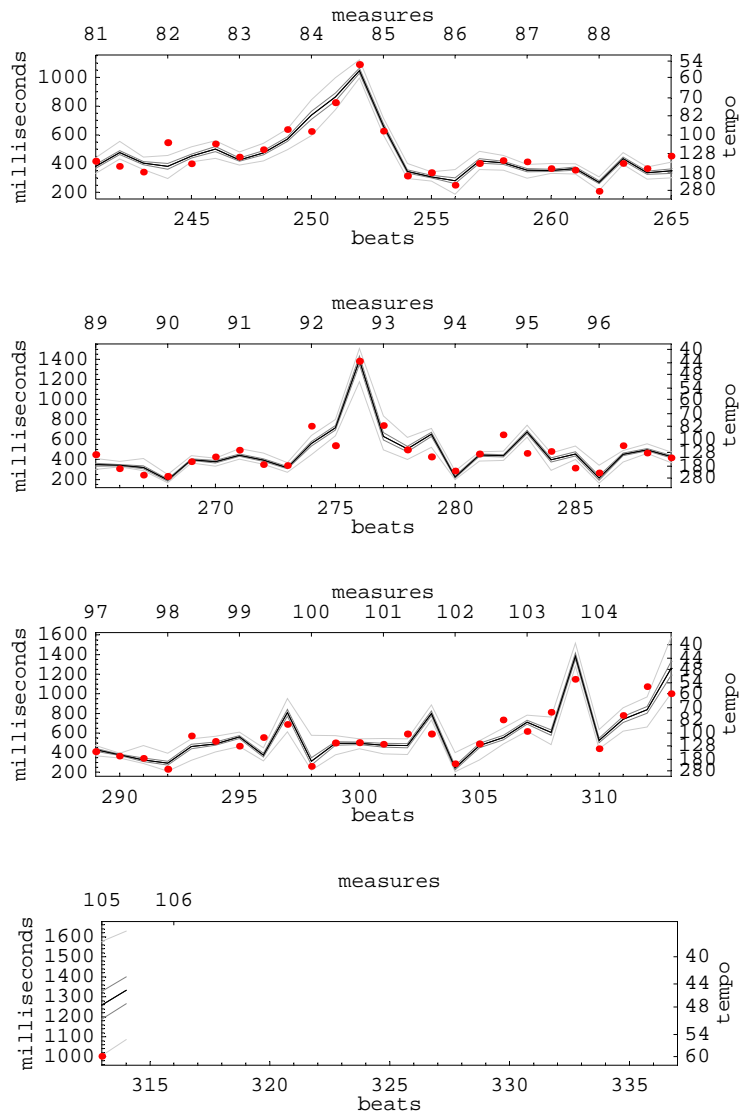
Here is a zoom in on the tempo for every 8 measures. The black line indicates the mean reverse-conducted durations for every beat. The dark gray lines surrounding the average duration line is the 95% confidence range for the true mean, and the light gray lines indicate the maximum and minimum durations for each beat from all reverse conducting trials.

The red dots indicates the beat event durations from the audio file which are more accurate than the average reverse conducting durations, and can be assumed to be the "correct" answer. Notice that the average reverse conducting beats are often one beat behind the red plot when the tempo changes suddenly which probably shows the delayed reaction time to listening to the performance.

```
Table[ShowFrame[newallplot, mindurs, maxdurs, 314, i, 24],  
{i, 0, Floor[318/8.0/3.0 + .99] - 1}];
```







Historical Recording Sample

Now do the same analysis on the historic recording sample, and compare any similarities/differences.

■ Loading the raw data

First load the absolute beat positions for the average beat positions and the corrected beat positions:

```
Get["pid5667230-10.corrected", Path → "D:\mazurkas\eval"]
mcbeats = pid5667230x10corrected;

Get["pid5667230-10.average", Path → "D:\mazurkas\eval"]
mabeats = Drop[pid5667230x10average, -1];
```


Now measure the time difference between each average and corrected beat. A negative value means that the average beat occurs before the corrected beat time.

```
Length[mabeats]
```

```
313
```

```
mdiffs = Transpose[mabeats][[2]] - Transpose[mcbeats][[2]]
```

```
{209, -1, 0, 51, 30, 62, 0, 0, 0, 29, 0, 0, 0, 0, 44, 81, 0, 0, 0, 49, 0, 0, 0, 0, 0, -184, 0,
 33, 100, 0, 96, 77, 67, 0, 0, 0, 0, 0, 33, 32, 94, 33, 0, -65, 44, 33, 0, 83, 0, -44, 30,
 90, 96, -16, 29, 33, 43, 0, 76, 0, 0, 36, 37, 38, 52, 13, 76, 172, 174, 184, -31, 110,
 19, 0, 0, 0, 55, 0, 15, 0, 0, 25, 13, 51, 0, 22, 0, 70, 96, 0, 58, 0, 0, 76, 89, 0, 89,
 122, 0, 57, 25, 0, 40, 100, 5, 108, 55, 64, 0, 0, 0, 73, 84, -37, 45, 113, 24, 59, 106,
 13, 51, 0, 0, 46, 0, 0, 44, 50, 18, 0, 16, 20, 29, 16, -12, 22, 30, 0, 0, 15, -21, 53,
 72, 35, 49, 19, 30, 28, 0, 0, 23, 36, 0, 0, 80, 37, 42, 22, 16, 43, 34, 13, 64, 0, -62,
 0, 0, 103, -71, -54, -157, 126, 200, 0, 71, 50, 58, 19, 36, 0, 0, 45, 39, 70, 124, 38,
 46, -28, -16, 0, -23, 83, -17, -30, -75, 38, 93, 55, 63, 0, 0, 0, 64, 0, 0, 0, -35,
 61, 24, 0, 0, -37, 0, 0, 0, 0, 17, 0, -241, 0, -14, 160, 116, 0, 70, 145, 0, 0, 62, 0,
 69, 120, 0, 0, 99, -144, -58, 41, 31, 0, 0, 0, 0, 61, -27, 83, 0, 0, 0, 23, 41, 74, 87,
 -203, 0, 59, 105, 38, 59, 77, 61, 0, 95, -20, 9, 47, 55, 52, 116, 72, 41, 39, 39, 68,
 0, 97, 0, -89, 33, 116, 149, 22, 31, 94, 54, 52, 130, 11, 40, 67, 87, 101, 133, 64,
 83, 51, 101, 91, 139, 18, 0, 0, 46, 26, 90, 71, -117, 0, 134, 122, -141, -290, -639}
```

The average correction to each beat from the average tap times of all trials was 46.2 milliseconds. This is almost 2 milliseconds better than the average for the modern recording sample.

```
<< Statistics`ConfidenceIntervals`
```

```
Mean[Abs[mdiffs]] // N
```

```
46.2684
```

■ Manual correction of average reverse conducting beat positions

This section displays the difference between the average reverse conducted beat positions and the values derived from manually correcting these values by ear using an soundfile editor.

This corrected beat duration data was measured by listening to the audio file and manually locating the beat positions in the soundfile. The initial positions of the average reverse conducting performance were used as a baseline, and this baseline was then adjusted so that it sounded as if all beats were occurring with the attacks of notes in the performance. There are probably a few errors in the corrected due to the tediousness of getting this data, and there are a few locations in the audio where the beat positions become vague, but overall, this data represents a highly accurate description of the pianists' actual beat-by-beat performance tempo, probably an accuracy within 10 milliseconds for all beats.

```
corrected = Transpose[mcbeats][[2]]

{2262, 2859, 3250, 3559, 3897, 4205, 4575, 4941, 5285, 5582, 5935, 6297, 6664, 7020,
 7306, 7579, 7956, 8318, 8615, 8870, 9242, 9624, 9997, 10386, 11103, 11860, 12188,
 12483, 12736, 13118, 13378, 13678, 13961, 14296, 14566, 14843, 15200, 15489, 15724,
 16025, 16245, 16583, 16930, 17280, 17505, 17804, 18154, 18512, 19505, 19960, 20270,
 20516, 20783, 21181, 21486, 21763, 22025, 22360, 22568, 22921, 23268, 23509, 23792,
 24080, 24331, 24675, 24932, 25186, 25462, 25697, 26146, 26427, 26883, 27268, 27625,
 28049, 28265, 28604, 28884, 29170, 29428, 29704, 29978, 30217, 30527, 30872, 31187,
 31397, 31619, 31959, 32207, 32526, 32801, 33026, 33258, 33613, 33857, 34084, 34466,
 34703, 34979, 35307, 35545, 35766, 36123, 36286, 36597, 36839, 37152, 37526, 37937,
 38181, 38412, 38798, 39043, 39252, 39609, 39855, 40062, 40409, 40705, 41036, 41363,
 41693, 42067, 42441, 42759, 43049, 43399, 43728, 43994, 44307, 44597, 44912, 45248,
 45533, 45864, 46246, 46569, 46906, 47272, 47543, 47824, 48192, 48491, 48825,
 49133, 49439, 49806, 50164, 50463, 50778, 51133, 51456, 51698, 52042, 52333, 52659,
 52989, 53267, 53607, 53965, 54222, 54607, 55039, 55422, 55880, 56337, 57093, 57662,
 58319, 58605, 58839, 59358, 59669, 60007, 60345, 60682, 60992, 61387, 61742, 62047,
 62382, 62634, 62876, 63252, 63526, 63877, 64212, 64573, 64951, 65227, 65639, 66011,
 66434, 66698, 66941, 67283, 67574, 67930, 68281, 68614, 68909, 69320, 69672, 70018,
 70396, 70615, 71006, 71411, 71778, 72170, 72533, 73005, 73452, 73862, 74473,
 74995, 75833, 76442, 77007, 77448, 77908, 78369, 78724, 79113, 79551, 79943,
 80254, 80678, 80995, 81285, 81686, 82030, 82292, 82831, 83111, 83335, 83652,
 83981, 84308, 84638, 84947, 85172, 85569, 85836, 86195, 86526, 86855, 87190,
 87511, 87907, 88599, 89405, 89741, 90029, 90270, 90636, 90923, 91187, 91486,
 91830, 92027, 92408, 92743, 92988, 93283, 93573, 93792, 94117, 94441, 94737,
 95033, 95309, 95697, 96134, 97342, 97956, 98260, 98513, 98779, 99176, 99484,
 99691, 99998, 100315, 100511, 100915, 101208, 101446, 101723, 101990, 102215,
 102569, 102861, 103170, 103408, 103681, 103874, 104250, 104585, 104847, 105094,
 105396, 105581, 105895, 106374, 106539, 106711, 106984, 107473, 107918, 108605}

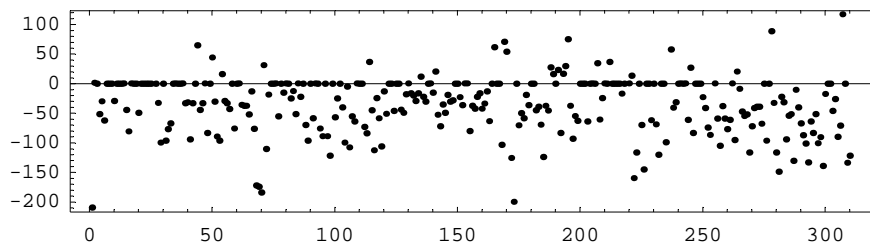
<< Statistics`ConfidenceIntervals`

avgtimes = Transpose[mabeats][[2]];
```

Here are the timing differences between the average reverse-conducting absolute times and the absolute times measured in a sound editor:

```
diffs = corrected - avgtimes;

ListPlot[diffs, Frame -> True, AspectRatio -> 1/4, PlotStyle -> PointSize[0.008]];
```

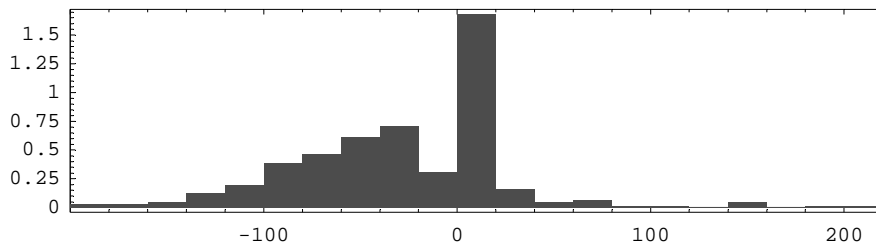


The average beats are noticeably not quite centered on the corrected beats. The average beats are, on the average, 27 milliseconds before the corrected beats. This is most likely due to the error in identifying the correct time offsets for each trial. 28 time points were used to align the trials to the soundfile. Probably the larger error in the offset value is due to the use of alignment beats from unstable tempo regions of the piece.

```
offseterror = Mean[diffs] // N
-27.1054
```

The histogram below shows the data from the plot above. It generally shows a nicely distributed range of corrections. The peak at 0 is due to the correction values being too small to change to the theoretically correct values about up to 10 milliseconds before their measured positions. Likewise, the distribution of corrections is offset from 0:

```
<< Graphics`Graphics`
Histogram[diffs, AspectRatio -> 1/4, Frame -> True, BarEdges -> False,
HistogramCategories -> 25, HistogramScale -> 100, HistogramRange -> {-200, 200}];
```



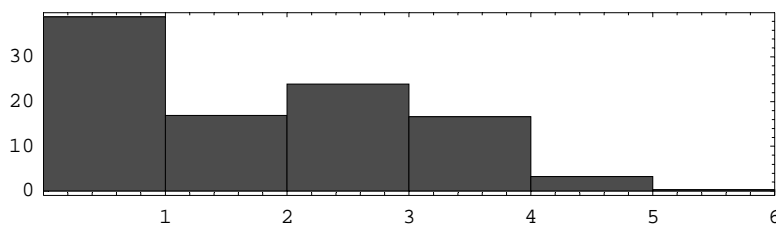
Similar to the modern sample, 1.5% of the average reverse conducting beats were sufficiently close to the manually audible analysis of the beat positions so that they did not require alteration to align with the audio beat events. The distribution should be smooth (since about 1/2 as much is expected to be exactly on the correct beat), so there are some beats marked as 0 ms difference to the true location which are occurring slightly after the actual beat position.

■ Trial Quality Measurements

Qualitatively, a score of 0-1 is an "A", 1-2 is a "B", 2-3 is a "C", 3-4 is a "D", 4-5 is an "E", and 5-6 is an "F". A score of 6 means the correction was 640 milliseconds, or about 2/3 of a second.

Mostly the corrections for the average of the tapping trials are better allocated between A, B, C, and D quality. Again, the "A" corrections were the most common. The worst case correction was between a value of 5 and 6 (for the last note in the piece). Other corrections in the range between 4 and 5 are also likely due to an unprepared change in tempo which was not expected beforehand.

```
<< Graphics`Graphics`
Histogram[Abs[QScore[mdiffs]], AspectRatio -> 1/4,
HistogramScale -> 100, Frame -> True, HistogramCategories -> 5];
```

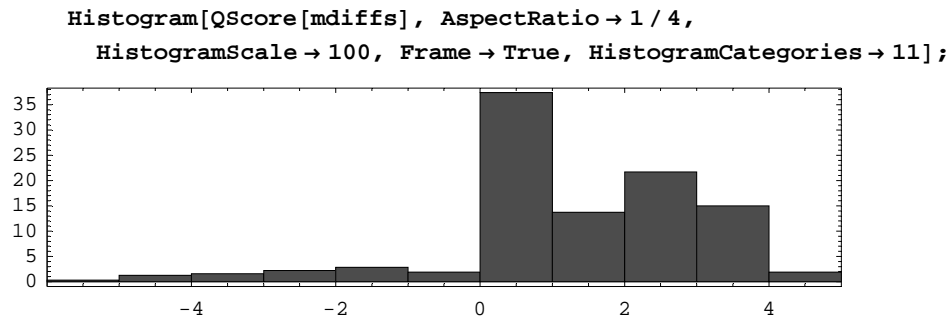


The range from 0-1 can be characterized as "no audible difference" (0-20 ms); 1-2 as "slight audible difference" (20-40 ms); 2-3 "minor audible difference" (40-80 ms); 3-4 as "noticeable audible difference" (80-160 ms), 4-5 as "very noticeable audible difference" (160-320 ms), and anything higher as "extremely noticeable audible difference" (> 320 ms).

35% of the average reverse conducting beats were at the level of no audible difference, with about 20% for each of the next three categories of "slight audible difference", "minor audible difference", and "noticeable audible difference". The higher number of beats in the "A" region may be partially due to the fact that the tempo of this performance is faster. The overall score for the average reverse conducting of the performance is a "B" (compared to a "B-" for the modern sample):

```
Mean[Abs[QScore[mdiffs]]]
1.64248
```

Here is a histogram display of the necessary corrections with positive and negative adjustments separated. Since the offset value was not estimated very accurately, it can be seen in the histogram that most average beats were occurring after the corrected beat than before it (or is that vice-versa?):



■ Examine the learning curve

How does the quality of the reverse conducting improve over time? Does the conductor learn to follow the performance better after repeated listening? First, load the individual tapping trials which contain precalculated offset values into the audio file of the performance recording:

```
Get["pid5667230-10-abs.ma", Path -> "D:\mazurkas\eval"]
data = pid5667230x10abs;
```

Now calculate the timing differences between the individual trials and the corrected beat times.

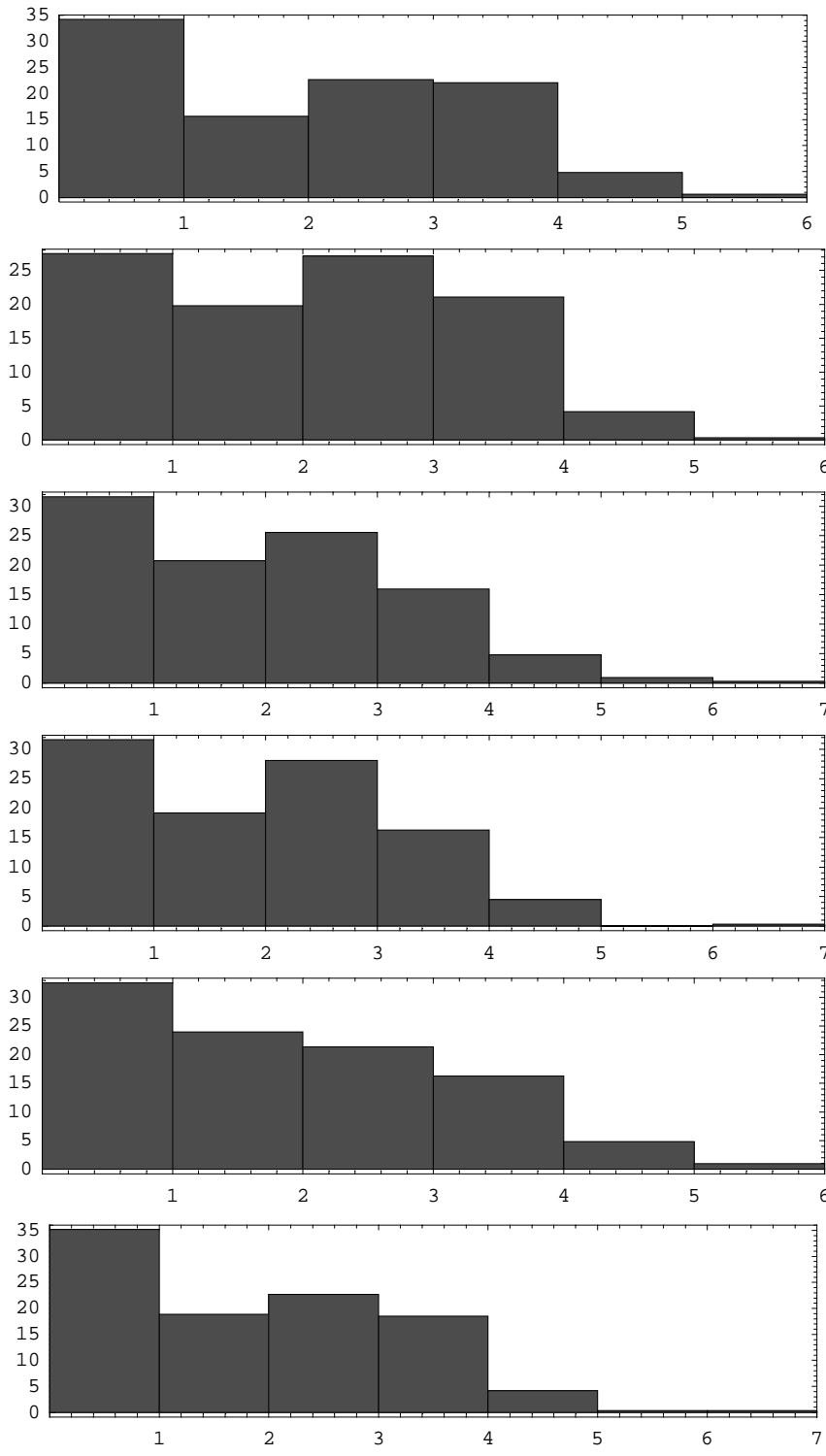
```
trialdiffs = Transpose[Drop[Transpose[data], -2] - Transpose[mcbeats][[2]]];
```

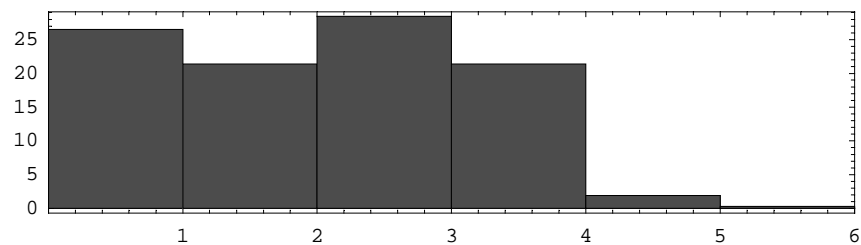
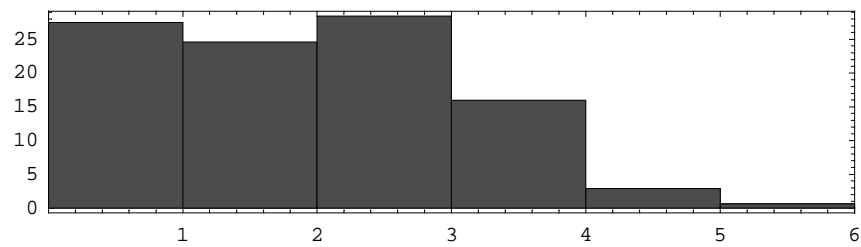
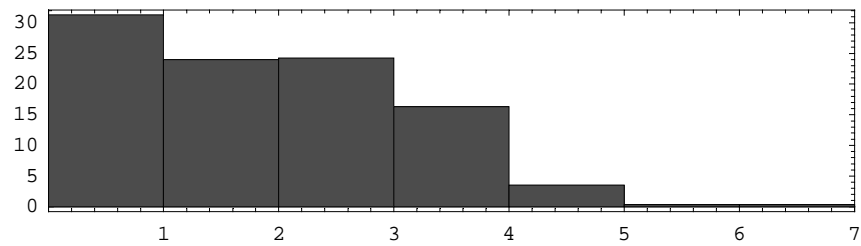
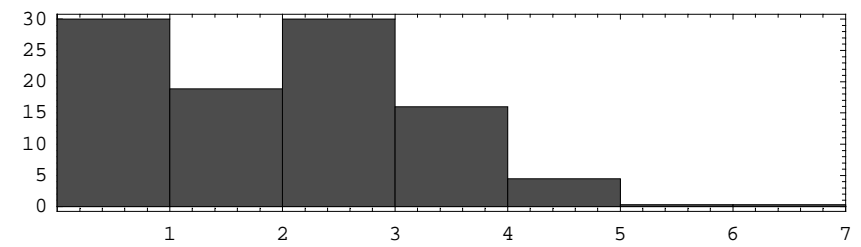
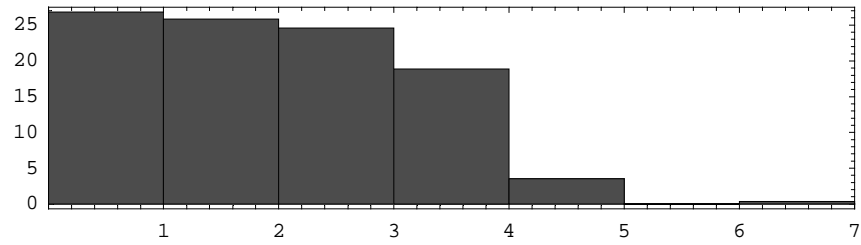
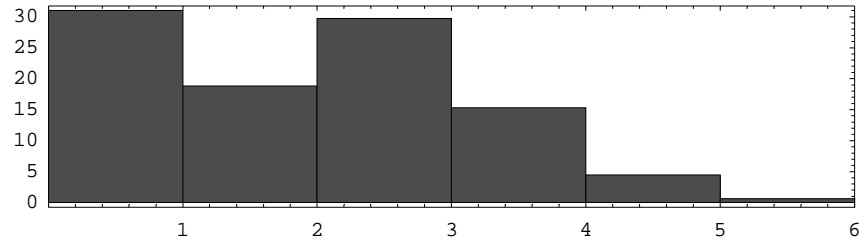
The first tapping trial gets a score of "B-", the worst trial gets a "C", and the best trial gets a "B-". The last trial is not the best trial.

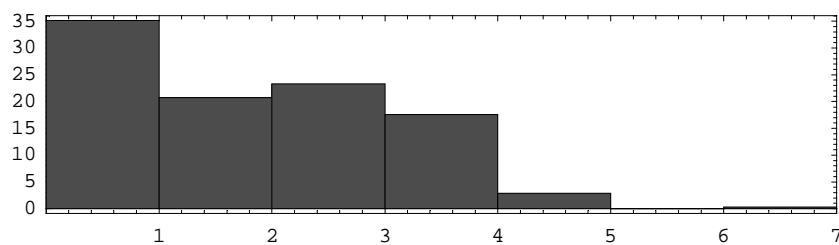
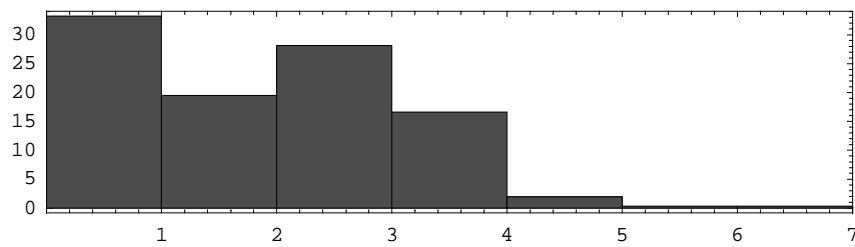
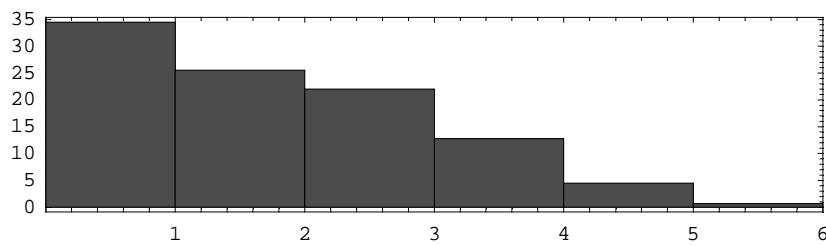
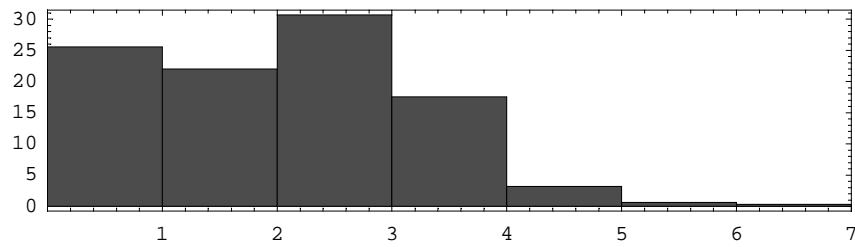
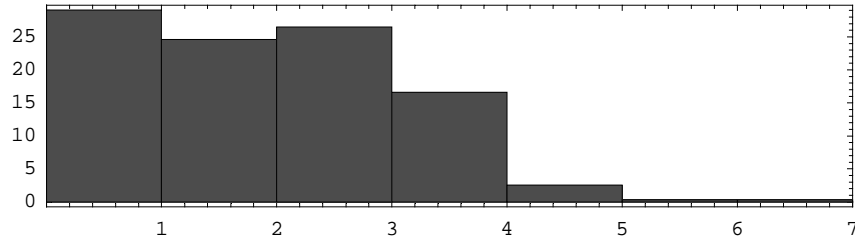
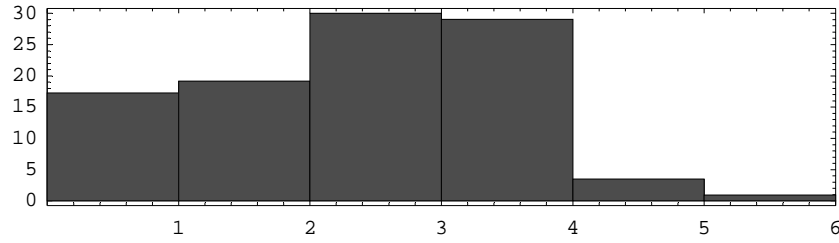
```
Map[Mean[Abs[QScore[#]]] &, trialdiffs]
{1.93812, 2.03309, 1.95114, 1.88829, 1.84084, 1.86074,
 1.92551, 1.96414, 1.90851, 1.85343, 1.88651, 1.96301, 2.2971,
 1.82462, 2.00065, 1.76715, 1.84735, 1.79655, 1.70999, 1.88778}
```

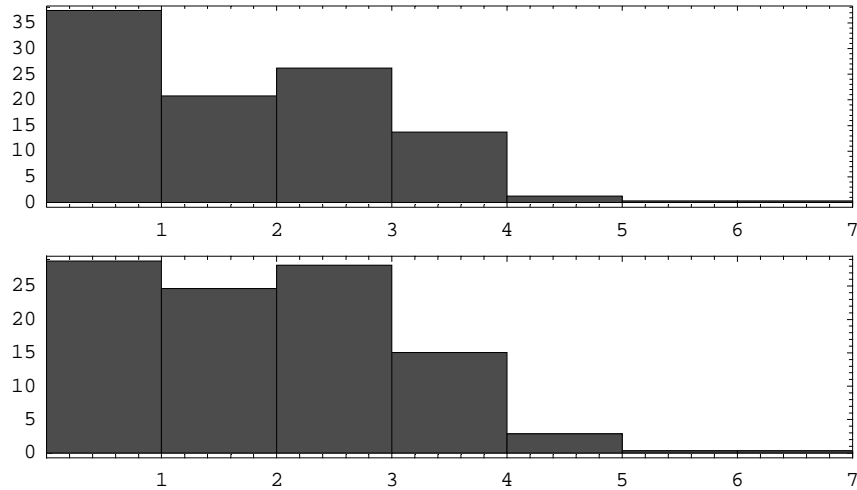
Here are quality score plots for each trial.

```
Table[Histogram[Abs[QScore[trialdiffs[[xz]]]], AspectRatio -> 1 / 4,
  Frame -> True, Range -> {{0, 6} {0, 30}}, HistogramScale -> 100,
  HistogramCategories -> 5], {xz, 1, Length[trialdiffs]}];
```









How does the accuracy of the first half of the trials compare to the second half of the trials?:

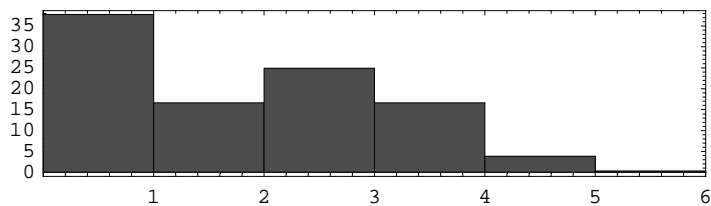
```

firsttrials = Drop[trialdiffs, -10];
lasttrials = Drop[trialdiffs, 10];

avgfirst = Map[Mean, Transpose[firsttrials]];
avglast = Map[Mean, Transpose[lasttrials]];

Histogram[Abs[QScore[avgfirst]], AspectRatio -> 1 / 4,
HistogramScale -> 100, Frame -> True, HistogramCategories -> 5];

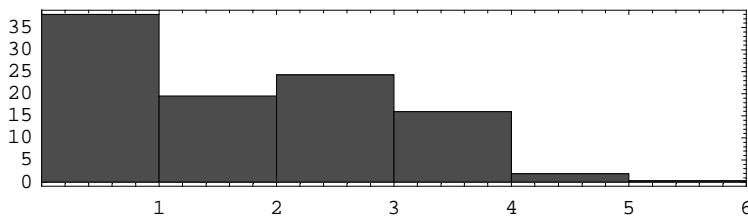
```



```

Histogram[Abs[QScore[avglast]], AspectRatio -> 1 / 4,
HistogramScale -> 100, Frame -> True, HistogramCategories -> 5];

```



So averaging only the last 10 trials of the set of 20 gives a slighter improvement in the average correction distance to the "true" beat compared to the modern sample.

The score both halves are on the opposite ends of the "B" range:

```
Mean[Abs[QScore[avgfirst]]]
```

```
1.72669
```



```
Mean[Abs[QScore[avglast]]]
1.67436
```

On the average, beats in the second half of the trials are 3.7 milliseconds closer to the "true" beat locations:

```
firsterr = Mean[Abs[avgfirst]]
50.05

lasterr = Mean[Abs[avglast]]
46.3063

firsterr - lasterr
3.74373
```

Also, averaging the performance of the second half of the trials does not improve the error displacement:

```
allerr = N[Mean[Abs[mdiffs]]]
46.2684

allerr - lasterr
-0.0379489
```

■ Plotting the average displacement error for each trial

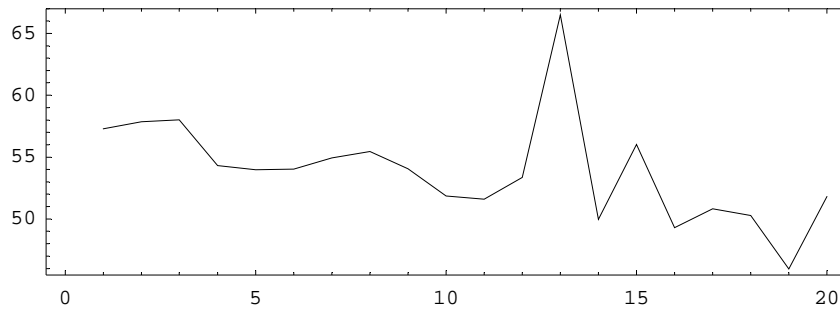
This section plots the average displacement error for each trial which shows the gradual improvement in the reverse conducting over time.

```
avgerr = Table[Apply[Plus, Sqrt[trialdiffs[[i]] * trialdiffs[[i]]]] /
  Length[trialdiffs[[i]]] * 1.0, {i, 1, Length[trialdiffs]}]

{57.2833, 57.8651, 58.0216, 54.3187, 53.9808, 54.0337,
 54.9245, 55.4393, 54.0576, 51.8491, 51.6096, 53.3464, 66.489,
 49.9617, 56.0288, 49.3032, 50.8296, 50.2775, 45.9693, 51.8}
```

Note that the average correction of the averaged data was 46 milliseconds, which is better than all of the individual trials except for #19.

```
errplot =
  ListPlot[avgerr, PlotJoined → True, Frame → True, AspectRatio → 1 / 3, Axes → False];
```

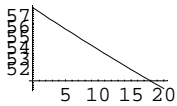


Now fit an exponentially decreasing line through the trial errors to model what the "learning curve" is for reverse conducting this performance.

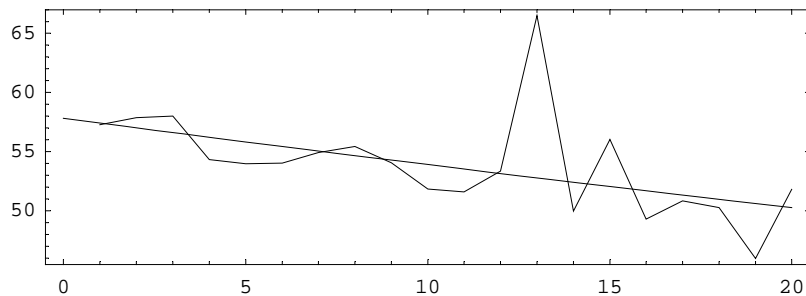
```
logfit = Fit[Log[10, avgerr], {1, x}, x]
```

```
1.762 - 0.00303853 x
```

```
fitplot = Plot[10^logfit, {x, 0, 20}];
```



```
Show[errplot, fitplot];
```



Predictions of accuracy can be derived from the learning curve over time, doubling the number of trials should result in an average error of 43 ms.

```
err40 = 10^logfit /. x → 40
```

```
43.6975
```

Doing 40 trials would improve the displacement error by about 2.6 milliseconds:

```
lasterr - err40
```

```
2.60882
```

After 100 trials, the accuracy would be about twice as good as after 20 trials:

```
err100 = 10^logfit /. x -> 100
28.7174
```

Doing 100 trials would improve the average displacement error by about 17.5 milliseconds (assuming an exponentially decaying learning curve):

```
lasterr - err100
17.589
```

■ Effect of removing the "bad" trial

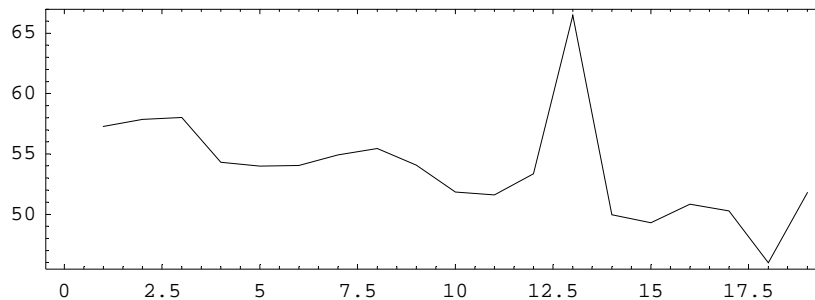
What happens if the "bad" trial (number 15 in this case) is removed from the average tapping time for each beat?

```
newerrset = RotateLeft[Drop[RotateRight[trialdiffs, 5], -1], 5];

newavgerr = Table[
  Apply[Plus, Sqrt[newerrset[[i]] * newerrset[[i]]] / Length[newerrset[[i]]] * 1.0 ,
  {i, 1, Length[newerrset]}]

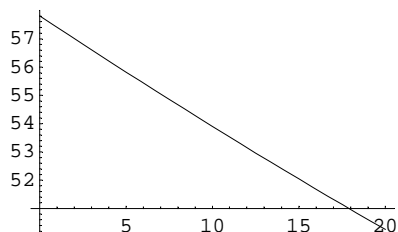
{57.2833, 57.8651, 58.0216, 54.3187, 53.9808, 54.0337, 54.9245, 55.4393, 54.0576,
  51.8491, 51.6096, 53.3464, 66.489, 49.9617, 49.3032, 50.8296, 50.2775, 45.9693, 51.8}

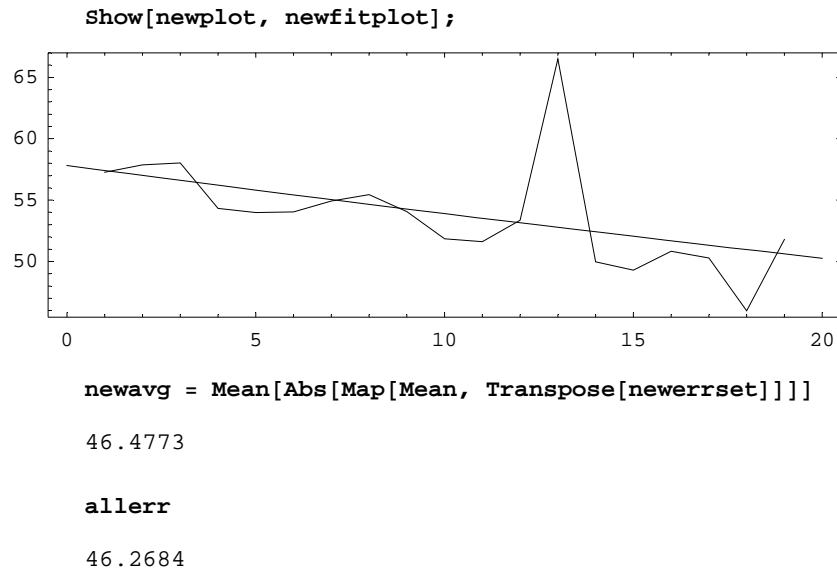
newplot =
  ListPlot[newavgerr, PlotJoined -> True, Frame -> True, AspectRatio -> 1 / 3, Axes -> False];
```



```
newlogfit = Fit[Log[10, newavgerr], {1, x}, x]
1.76315 - 0.0034016 x
```

```
newfitplot = Plot[10^logfit, {x, 0, 20}];
```

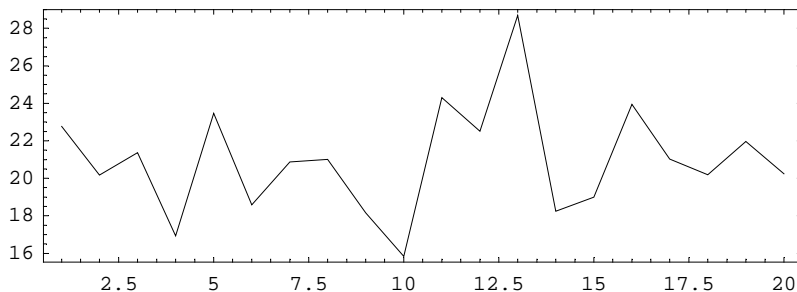
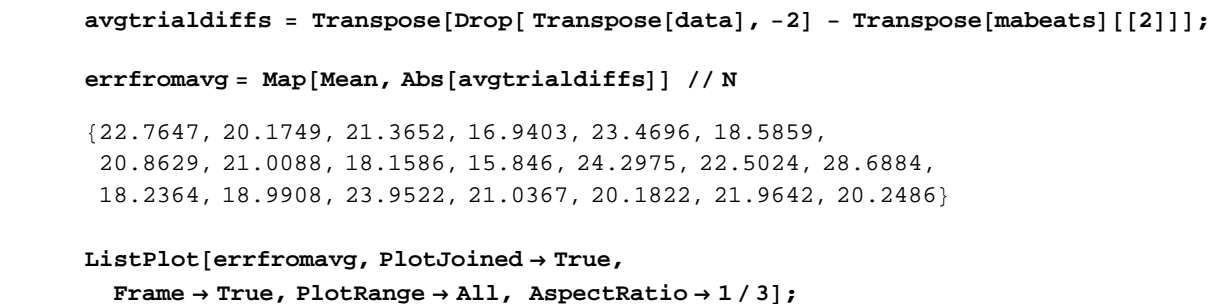




Removing the "bad" trial decreases the overall average displacement error in this case by about 0.2 milliseconds which is very small.

■ Trying to identifying "bad" performances from the average

Instead of comparing the individual performances to the corrected data, compare them to the average of all trials.



Trial 13 is still identified as the "worst" performance when compared to the average of all trials rather than to the corrected times. So if it is necessary to throw out "bad" performances, then this method would work in identifying the bad performances.

■ Dropping any single trial

Dropping the worst trial does not improve the accuracy of the average. Does dropping any other trial improve the average of all trials?

```
droppedsets = Table[RotateRight[Drop[RotateLeft[trialdiffs, i], -1], i], {i, 1, 20}];

droppedavg = Table[Mean[Abs[Map[Mean, Transpose[droppedsets[[i]] ]]]], {i, 1, 20}]

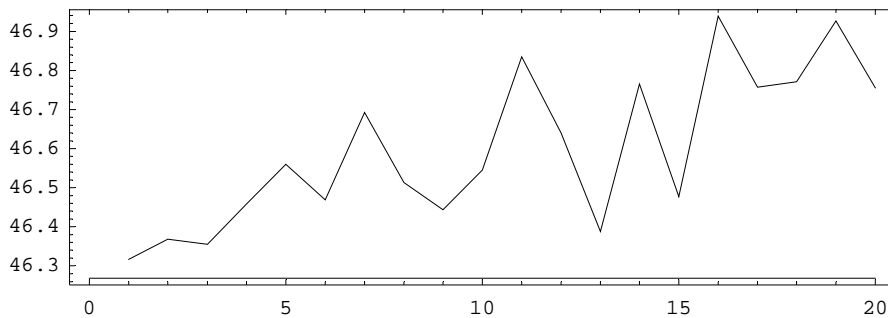
{46.3173, 46.3685, 46.3553, 46.4583, 46.5598, 46.4696,
 46.6924, 46.5135, 46.4436, 46.5452, 46.8349, 46.6399, 46.3883,
 46.7653, 46.4773, 46.9385, 46.7573, 46.7708, 46.9269, 46.7556}

plot = ListPlot[droppedavg, PlotJoined → True,
  Frame → True, Axes → False, DisplayFunction → Identity];

allerr

46.2684

Show[plot, Graphics[Line[{{0, allerr}, {20, allerr}}]],
  DisplayFunction → $DisplayFunction, AspectRatio → 1 / 3];
```



If any one trials is excluded, then the average displacement error is increased from the average of the rest of the trials. The horizontal line at the bottom of the plot represents the displacement error when averaging all trials.

■ Dropping a range of trials

Now examine how the displacement error changes as more and more of the earlier trials are removed from the average.

```
dsets = Table[Drop[trialdiffs, i], {i, 1, 19}];

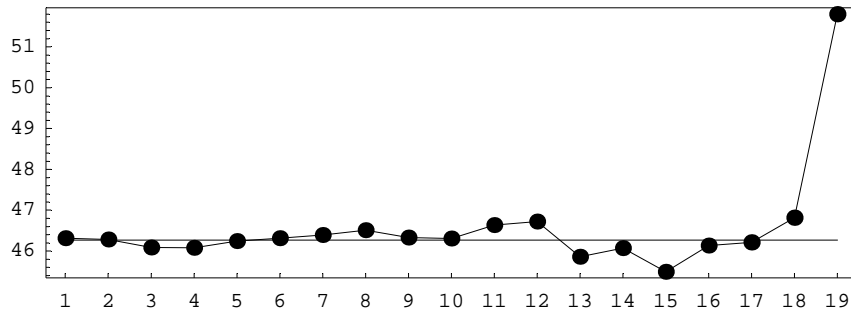
davg = Table[Mean[Abs[Map[Mean, Transpose[dsets[[i]] ]]]], {i, 1, 19}]

{46.3173, 46.279, 46.084, 46.0804, 46.2452, 46.3173, 46.3951, 46.5077, 46.3309, 46.3063,
 46.6345, 46.7263, 45.8615, 46.074, 45.4964, 46.1338, 46.2111, 46.8121, 51.8}

dplot = ListPlot[davg, PlotJoined → True,
  Frame → True, Axes → False, DisplayFunction → Identity];
```

```
dplotp = ListPlot[davg, PlotStyle → PointSize[0.02],
  Frame → True, Axes → False, DisplayFunction → Identity];

dropstart = Show[dplot, dplotp, Graphics[Line[{{1, allerr}, {19, allerr}}]],
  FrameTicks → {Range[19], Automatic, None, None},
  DisplayFunction → $DisplayFunction, PlotRange → All, AspectRatio → 1 / 3];
```



Dropping the first 15 trials gives the best error of under 46 ms, which is 0.5 ms better than averaging all trials together. Dropping the first 10 trials yields the same average error as including all 20 trials.

Just for fun, here is a plot created by dropping the later trials more and more:

```
rdsets = Table[Drop[trialdiffs, -i], {i, 1, 19}];

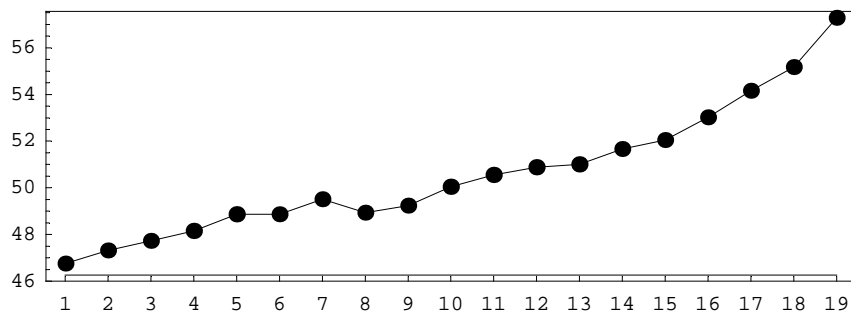
rdavg = Table[Mean[Abs[Map[Mean, Transpose[rdsets[[i]]]]]], {i, 1, 19}]

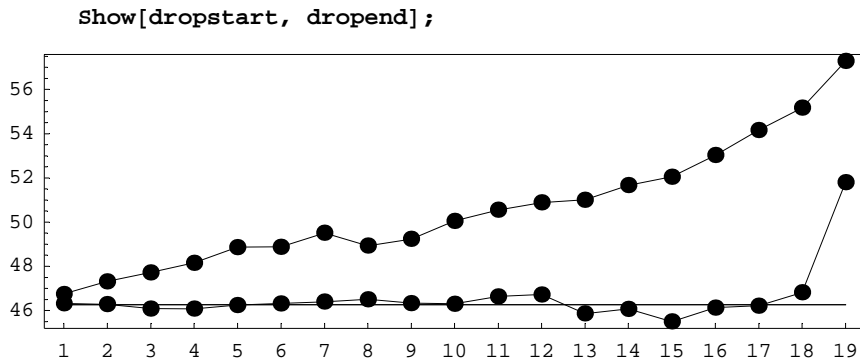
{46.7556, 47.3185, 47.7238, 48.1523, 48.8699, 48.8746, 49.5048, 48.9287, 49.2403, 50.05,
  50.5512, 50.8827, 51.0014, 51.6678, 52.0483, 53.0239, 54.1639, 55.1734, 57.2833}

rdplotp = ListPlot[rdavg, PlotStyle → PointSize[0.02],
  Frame → True, Axes → False, DisplayFunction → Identity];

rdplot = ListPlot[rdavg, PlotJoined → True,
  Frame → True, Axes → False, DisplayFunction → Identity];

dropend = Show[rdplot, rdplotp, Graphics[Line[{{1, allerr}, {19, allerr}}]],
  FrameTicks → {Range[19], Automatic, None, None},
  DisplayFunction → $DisplayFunction, PlotRange → All, AspectRatio → 1 / 3];
```





You definitely do not want to drop later trials from the average, since dropping any later trials and keeping the earlier trials will always significantly increase the average displacement error.

Also note that all individual trials are worse than trial average (except trial #19), as well as averages involving only the later trials:

```

individ = Map[Mean[N[Abs[Drop[#, -2] - Transpose[mcbeats][[2]]]]] &, data]

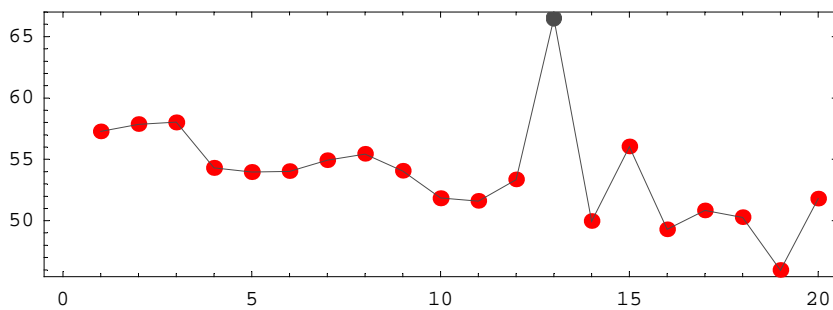
{57.2833, 57.8651, 58.0216, 54.3187, 53.9808, 54.0337,
 54.9245, 55.4393, 54.0576, 51.8491, 51.6096, 53.3464, 66.489,
 49.9617, 56.0288, 49.3032, 50.8296, 50.2775, 45.9693, 51.8}

individplot = ListPlot[individ, PlotStyle -> {Hue[0.0], PointSize[0.02]},
  Frame -> True, Axes -> False, DisplayFunction -> Identity, AspectRatio -> 1 / 3];

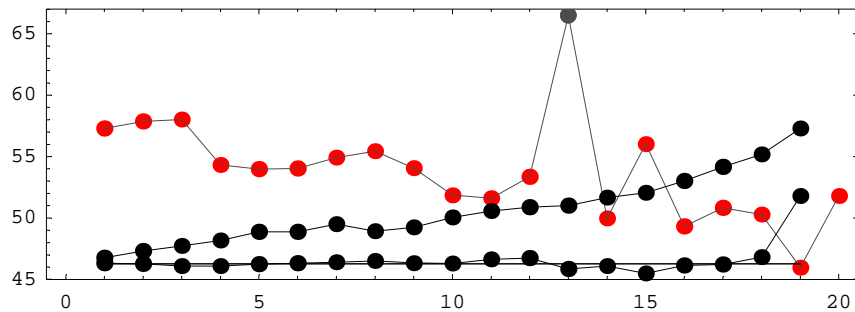
individplotp = ListPlot[individ, PlotJoined -> True, Frame -> True,
  Axes -> False, PlotStyle -> Hue[0.0], DisplayFunction -> Identity];

Show[individplot, individplotp, DisplayFunction -> $DisplayFunction];

```



```
Show[individplot, individplotp,
      dropstart, dropend, DisplayFunction -> $DisplayFunction];
```



The red curve in the plot above shows the average displacement errors from the corrected times for each of the 20 individual trials. Only trials 14 and 20 are close to the error rate of the average of all trials.

■ Offset sensitivity

Individual trials were aligned to the Modern test case using a few points which seemed to be in a stable tempo region. How accurate is the offset calculated from these few points in the audio? Would the average of all trials improve if an offset is calculated for each trial based on the corrected data?

```
SquareDiff[x_, y_] := Apply[Plus, (x - y) * (x - y)]

AlignSequences[seq1_, seq2_] := Module[
  {diffs},
  diffs = Table[{x, SquareDiff[seq1, seq2 + x]}, {x, -10000, 10000, 100}];
  (x /. Solve[D[Fit[diffs, {1, x, x^2}, x], x] == 0, x])[1]
]

AlignSequences[{1, 2, 3}, {2, 3, 4}]

-1.

addoffsets = Table[AlignSequences[Transpose[mcbeats][[2]], Drop[data[[i]], -2]],
  {i, 1, Length[data] - 2}]

{-34.1719, -33.0301, -27.2827, -25.7365, -27.1949, -26.0596,
 -26.0757, -32.0255, -25.5961, -27.5199, -14.5126, -34.9042,
 -42.8652, -21.5527, -31.8552, -16.7703, -23.1103, -23.2301}

newmabeats = Drop[Map[Mean, Transpose[Drop[data, -2] + addoffsets]], -2];

extraoffset = Mean[addoffsets]

-27.4163
```



```

Min[addoffsets]
Max[addoffsets]

-42.8652

-14.5126

```

Using more accurate offsets, the average displacement error of the average of all trials would be about 2 ms better:

```

omdiffs = Transpose[mcbeats][[2]] - newmabeats;

N[Mean[Abs[omdiffs]]]

44.1637

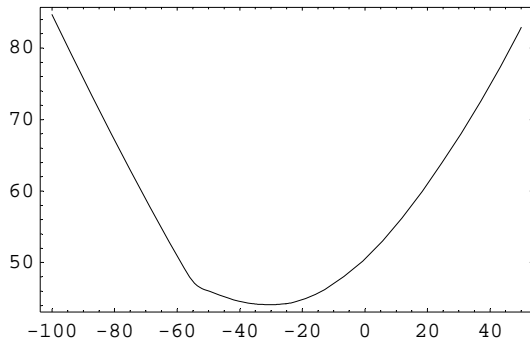
```

Don't know what it means, but the following plot shows that there is a wider minimum in the error rate when shifting the averaged trial differences when the offsets have been optimized for all beats in the piece, rather than a small set as shown in the second plot:

```

Plot[N[Mean[Abs[omdiffs + x - Mean[addoffsets] ]]],
{x, -100, 50}, Frame → True, Axes → False];

```

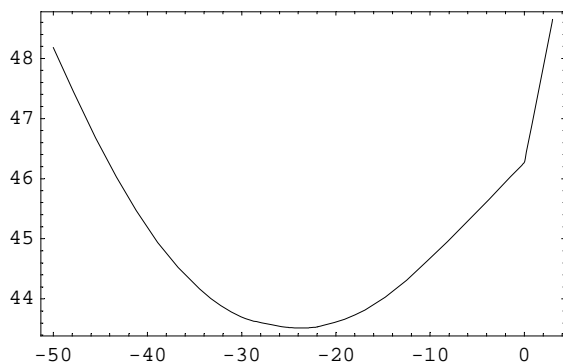


Here is a plot using the initial offset values:

```

Plot[N[Mean[Abs[mdiffs + x ]]], {x, -50, 3}, Frame → True, Axes → False];

```



■ Comparing corrected beats to average reverse conducting beats

This section plots the average tempo range and compares it to the actual tempos measured from the audio file.

```

Get["pid5667230-10.ma", Path → "D:\mazurkas\eval"]
data = Transpose[pid5667230x10];

eachbeatstats = Map[DispersionReport, data] // N;

confidence =
  Map[MeanCI[#[[1]], KnownStandardDeviation → (KnownStandardDeviation /. #[[2]])] &,
    Transpose[{data, eachbeatstats}]];

durationmean = Map[Mean, data] // N;

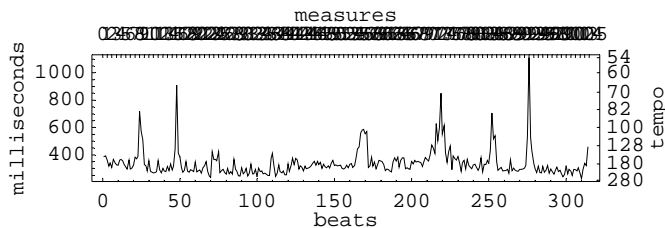
maxdurs = Map[Max, data];

mindurs = Map[Min, data];

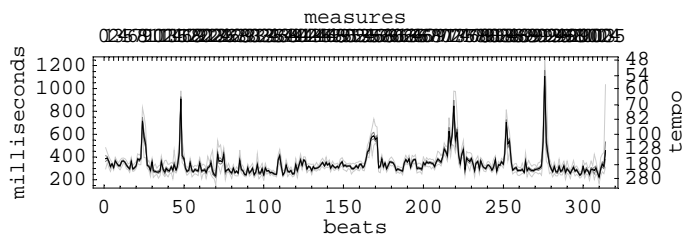
meanhighplot = ListPlot[Transpose[confidence][[1]], PlotJoined → True,
  DisplayFunction → Identity, PlotStyle → RGBColor[.5, .5, .5]];
meanlowplot = ListPlot[Transpose[confidence][[2]], PlotJoined → True,
  DisplayFunction → Identity, PlotStyle → RGBColor[.5, .5, .5]];
maxplot = ListPlot[maxdurs, DisplayFunction → Identity,
  PlotJoined → True, PlotStyle → RGBColor[0.8, 0.8, 0.8]];
minplot = ListPlot[mindurs, DisplayFunction → Identity,
  PlotJoined → True, PlotStyle → RGBColor[0.8, 0.8, 0.8]];

meanplot = ListPlot[durationmean,
  PlotJoined → True, PlotRange → All, AspectRatio → 1/4, Frame → True,
  FrameLabel → {"beats", "milliseconds", "measures", "tempo"},
  FrameTicks → {Automatic, Automatic, Table[{x, x/3}, {x, 0, 315, 3}],
    Map[{60000.0 / #, #} &, {30, 40, 44, 48, 54, 60, 70, 82, 100, 128, 180, 280}]}];

```

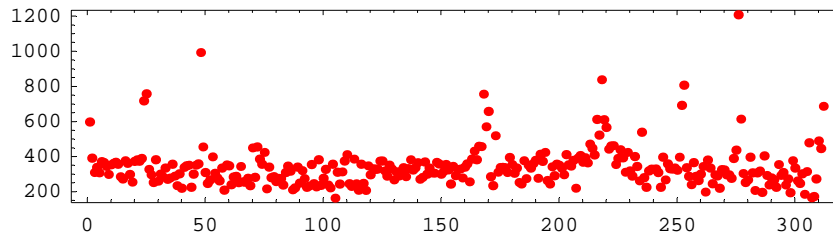


```
allplot = Show[meanplot, maxplot, minplot, meanhighplot, meanlowplot, meanplot];
```

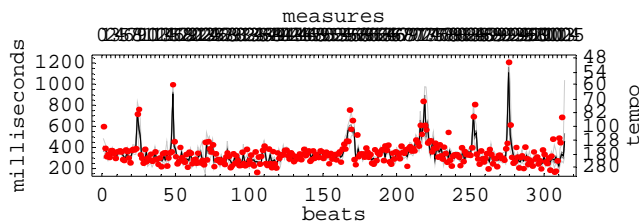


```
correctdeltas = Drop[RotateLeft[corrected, 1] - corrected, -1];
```

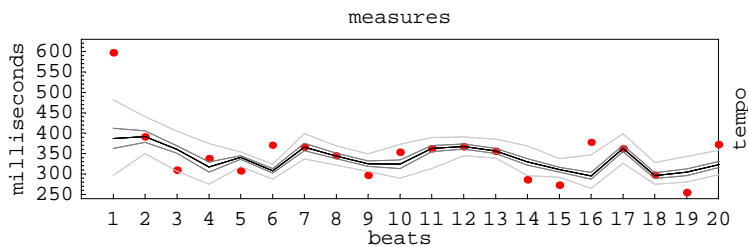
```
correctplot = ListPlot[correctdeltas, PlotStyle -> {Hue[0.0], PointSize[0.012]},
  PlotRange -> All, Frame -> True, AspectRatio -> 1/4];
```



```
newallplot = Show[allplot, correctplot];
```



```
finalplot = Show[allplot, correctplot, allplot, Frame -> True,
  PlotRange -> {{0, 20}, {240, 630}}, FrameTicks -> {Range[20], Automatic, None, None}];
```



```
ShowFrame[finalplot_, minrange_, maxrange_, len_, frame_, width_] := Module[
  {framecount, startpoint, endpoint, rmin, rmax, starti, endi, delta},
  framecount = len / width + 0.99 // Floor;
  startpoint = frame * width + 1;
  endpoint = (frame + 1) * width + 1;
  starti = (startpoint - 1) + 1;
  endi = (endpoint - 1) + 1;
  If[endi > len, endi = len];
  If[starti > len, starti = len];
  rmax = Max[Take[maxrange, {starti, endi}]];
  delta = rmax * 0.03;
  rmax = rmax + delta;
  rmin = Min[Take[minrange, {starti, endi}]];
  rmin = rmin - delta;
  Show[finalplot, PlotRange -> {{startpoint, endpoint}, {rmin, rmax}},
    FrameLabel -> {"beats", "milliseconds", "measures", "tempo"},
    FrameTicks -> {Automatic, Automatic, Table[{x + 1, x / 3 + 1}, {x, 0, 315, 3}],
      Map[{60000.0 / #, #} &, {30, 40, 44, 48, 54, 60, 70, 82, 100, 128, 180, 280}]},
    Axes -> False]
]
```

Here is a zoom in on the tempo for every 8 measures. The black line indicates the mean reverse-conducted durations for every beat. The dark gray lines surrounding the average duration line is the 95% confidence range for the true mean, and the light gray lines indicate the maximum and minimum durations for each beat from all reverse conducting trials.

The red dots indicates the beat event durations from the audio file which are more accurate than the average reverse conducting durations, and can be assumed to be the "correct" answer. Notice that the average reverse conducting beats are often one beat behind the red plot when the tempo changes suddenly which probably shows the delayed reaction time to listening to the performance.

```
Table[ShowFrame[newallplot, mindurs, maxdurs, 314, i, 24],
      {i, 0, Floor[318/8.0/3.0+.99] - 1}];
```

