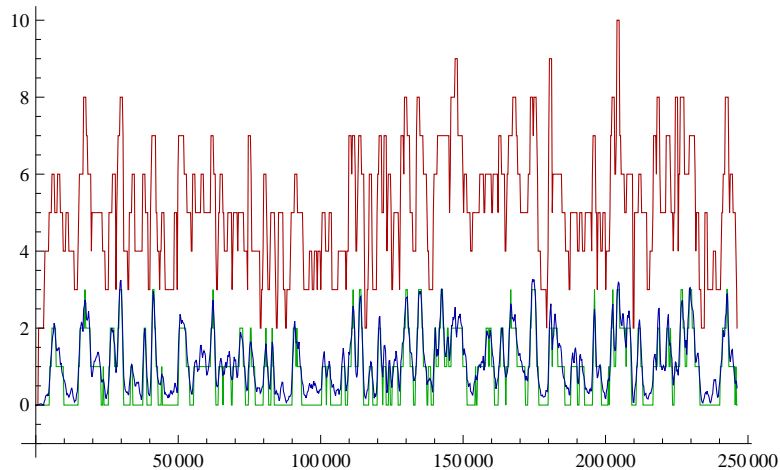


Mazz dirty colormap tile test - entire mesh, game tick resolution

1. Dirty tiles vs Game Time: Rolling Max, Median, Mean (R,G,B)

```
radius = 500; counts = Normal@SparseArray[First /@ data → Last /@ data];  
ListPlot[{MaxFilter[#, radius], MedianFilter[#, radius], MeanFilter[#, radius]} &@counts, PlotRange → All,  
AxesOrigin → {0, -1}, Joined → True, MaxPlotPoints → 1000, PlotStyle → Darker@{Red, Green, Blue}]
```

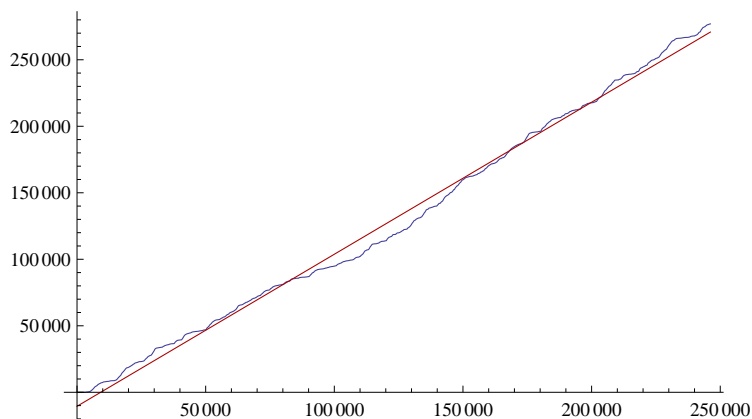


2. Cumulative dirty tiles vs Game Time

```
cumdata = FoldList[{{#2[[1]], #1[[2]] + #2[[2]]} &, {0, 0}, data];  
linefit = Fit[cumdata, {1, t}, t]
```

-10 507.6 + 1.14289 t

```
Show[ListPlot[cumdata, PlotRange → All, Joined → True, MaxPlotPoints → 1000],  
Plot[linefit, {t, 0, First@Last@data}, PlotStyle → Darker@Red]]
```



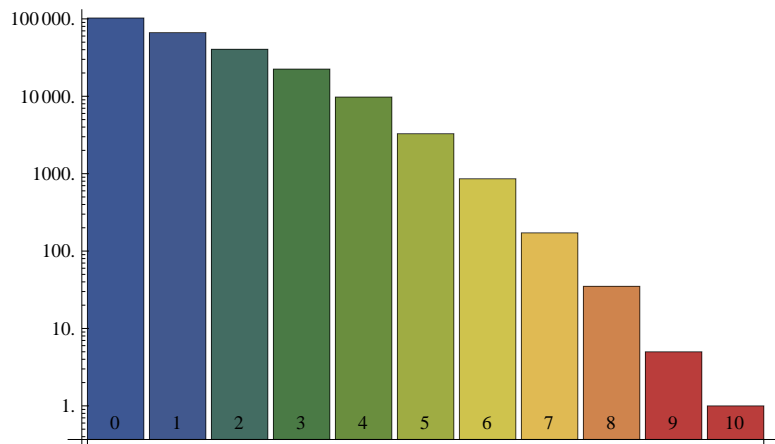
2b. Lower bound for bandwidth savings (percent) over entire game assuming 13-20 visible tiles

```
100 (1 - Coefficient[linefit, t] / {13, 20})
```

{91.2085, 94.2856}

3. Dirty tile count distribution

```
bars = Join[{{0, First[Last[data]] - Length[data]}}, Sort@Tally[Last /@ data]];
BarChart[Last /@ bars, ChartStyle -> "DarkRainbow", ChartLabels -> First /@ bars, ScalingFunctions -> "Log"]
```



4. Distribution of k-spike runs (groups of consecutive ticks where # dirty tiles > k)

```
TableForm@Partition[#, 3] && Table[
  Labeled[ListLogLogPlot[Sort@Tally[Length /@ Select[Split@Thread[Last /@ data > k], First]],
    Joined -> True, PlotRange -> All, PlotLabel -> "k=" <> ToString[k],
    AxesLabel -> {"", "frequency"}], "run length"], {k, 1, 9}]
```

