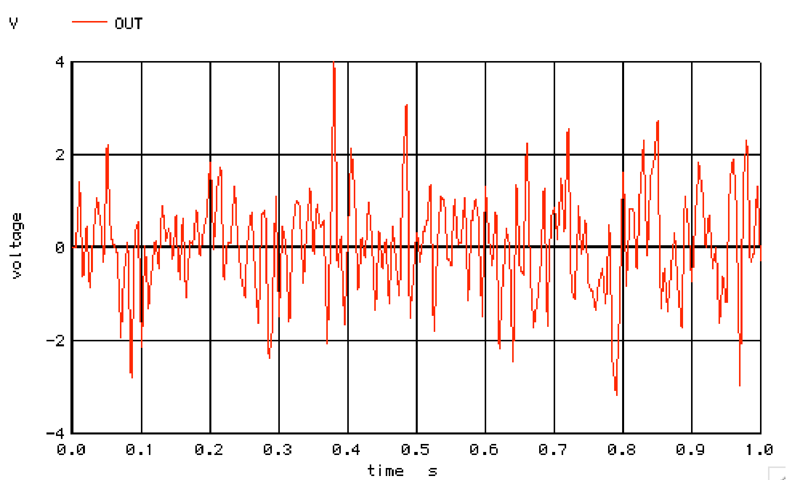
===========NYQUIST_LIMITED_NOISE=========================

**What does random noise look like at a lower sample rate?
This time a noise file is created which has one tenth the bandwidth.**

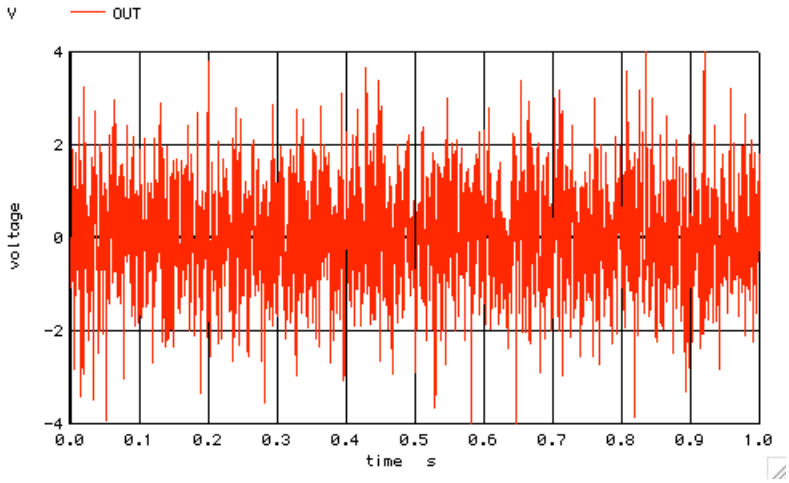
```
MacSpice 2 -> rndsrc 5m   1
PWL_File.inc has been created in the MacSpice folder
MacSpice 3 ->
```

```
PWL_Noise_1VRMS_@100Hz
* www.idea2ic.com
* dsauersanjose@aol.com  2.11.10_10.49AM   replace(OPT-SPACE)=>SPACE
*     OUT     Rload
*      _____/\  __           need to have a file called
*     _|_     \/  |          "rndsrc" in the "MacSpice"
*    /   \        |          folder inside "Doucments"
*  /VpwlT\        |
*  \     /       _|_         First typing in a MacSpice
*   \___/        ///         window "rndsrc  5m 1"
*     |          Gnd
*    _|_                     Then run this file
*    ///
*    Gnd
* timestep =  5m      means 100 bandwidth
* duration = 1        means 1Hz resolution
* type into a         MacSpice window =>  "rndsrc .5m 1"
* it will generate    PWL_File.inc in this format..
* VpwlT OUT 0         PWL( + 0.005 0.988835 +.....
*===========Circuit_Netlist=========================
.include    PWL_File.inc
Rload       OUT     0        1k
*TRAN       TSTEP   TSTOP    TSTART  TMAX        ?UIC?
.tran       .5m     1        0       .5m        UIC
*===========Run_Transient==========================
.control
run
set         pensize =  1
plot        OUT     ylimit -4 +4
*=================================================================
```

**It turns out that the PWL file that is produced
will have the same RMS value, but with one tenth
the bandwidth.**



**The new 100Hz random noise is in fact equivalent
to taking every tenth sample from a 1Khz noise
file. That actually does not effect the RMS value.**




```
*===========Find_RMS_Input=========================
let num =    length(out)-1
let i  =     0
let vpwr =   0
repeat       $&num
let i =      i +1
let vpwr =   vpwr  + (mag(OUT[i])*mag(OUT[i]))/num
end
let vrms1 = sqrt(vpwr)
echo         INPUT RMS =  $&vrms1
*==================================================================
```

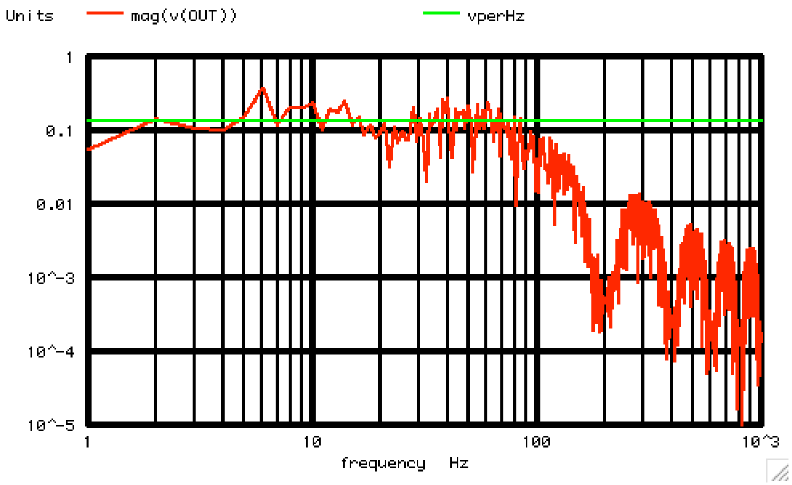
Performing a real RMS on the input can show this is true..

Circuit: PWL_Noise_1VRMS_@1KHz

INPUT RMS = 1.00617

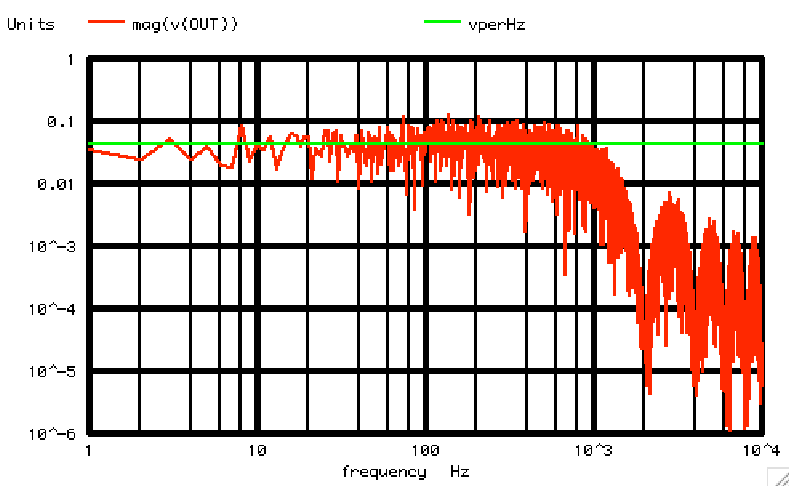
```
*===========Find_Spectrum==========================
linearize
set          specwindow =    "rectangular"
*SPEC        FSTART  FSTOP    FSTEP   VECTOR
spec         1       1000     1       v(OUT)
*===========Find_Spectrum_RMS======================
let num =    length(out)-1
let i  =     0
let vpwr =   0
repeat       $&num
let i =      i +1
let vpwr =   vpwr  + mag(OUT[i])* mag(OUT[i])
end
let vrms2 = sqrt(vpwr)
echo         SPECTRUM RMS =  $&vrms2
*===========View_Spectrum==========================
let          BandW = 100
let          vperHz = vrms2/sqrt(BandW)
set          pensize       = 2
plot         mag(v(OUT)) vperHz  loglog
.endc
.end
*==================================================================
```

But now the same 1Vrms of noise is now packaged into 100Hz of Bandwidth.



SPECTRUM RMS = 1.35911

This is just the classical case of not using an anti-aliasing filter. The 100Hz spectrum is shown above. The 1kHz below. The 100Hz noise is like taking every one tenth sample from the 1KHz noise. By doing so, the 100Hz to 1KHz noise just gets aliased into the 100Hz noise spectrum.



SPECTRUM RMS = 1.36041

One does not need a lot of data points to find the standard deviation of something. Adding more data points does not really change a standard deviation. It just makes the rms or standard deviation value more precise.

Provided one's sampler is fast enough, under sampling noise provides the full noise under its full bandwidth at its full standard deviation or RMS value.

```
===========Full_Netlist_For_Copy_Paste=======================
PWL_Noise_1VRMS_@100Hz
* www.idea2ic.com
* dsauersanjose@aol.com  2.11.10_10.49AM   replace(OPT-SPACE)=>SPACE
*    OUT     Rload
*     _____/\ __        need to have a file called
*   _|_     \/  |       "rndsrc" in the "MacSpice"
*  / ‾ \        |       folder inside "Doucments"
* /VpwlT\       |
```

```
* \     /        _|_      First typing in a MacSpice
*  \___/        7//7      window "rndsrc  5m 1"
*      |        Gnd
*      |                  Then run this file
*    7//7
*    Gnd
* timestep =  5m      means 100 bandwidth
* duration = 1        means 1Hz resolution
* type into a         MacSpice window =>  "rndsrc .5m 1"
* it will generate    PWL_File.inc in this format..
* VpwlT OUT 0         PWL( + 0.005 0.988835 +.....
*===========Circuit_Netlist=========================
.include    PWL_File.inc
Rload       OUT     0         1k
*TRAN       TSTEP   TSTOP     TSTART  TMAX         ?UIC?
.tran       .5m     1         0       .5m         UIC
*===========Run_Transient=========================
.control
run
set         pensize =  1
plot        OUT     ylimit -4 +4
*===========Find_RMS_Input=========================
let num =   length(out)-1
let i  =    0
let vpwr =  0
repeat      $&num
let i =     i +1
let vpwr =  vpwr  + (mag(OUT[i])*mag(OUT[i]))/num
end
let vrms1 = sqrt(vpwr)
echo        INPUT RMS =  $&vrms1
*===========Find_Spectrum=========================
linearize
set         specwindow =    "rectangular"
*SPEC       FSTART  FSTOP    FSTEP   VECTOR
spec        1       1000      1       v(OUT)
*===========Find_Spectrum_RMS=========================
let num =   length(out)-1
let i  =    0
let vpwr =  0
repeat      $&num
let i =     i +1
let vpwr =  vpwr  + mag(OUT[i])* mag(OUT[i])
end
let vrms2 = sqrt(vpwr)
echo        SPECTRUM RMS =  $&vrms2
*===========View_Spectrum=========================
let         BandW = 100
let         vperHz = vrms2/sqrt(BandW)
set         pensize       = 2
plot        mag(v(OUT)) vperHz  loglog
.endc
.end

rndsrc 5m   1

2.12.10_2.44PM
dsauersanjose@aol.com
Don Sauer
http://www.idea2ic.com/
```