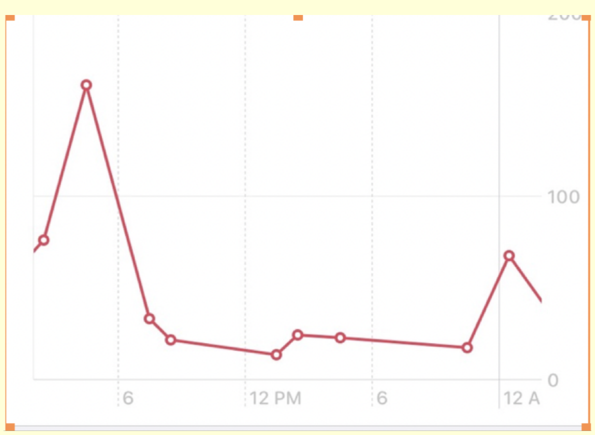
Heart Rate Variability & Correct Measurement

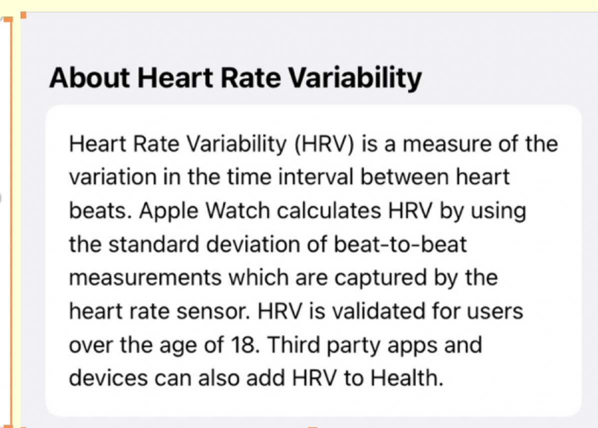
Summary: HRV appears to be ill defined in papers and regressions; using logged variables fixes the problem.

Introduction

Antifragile discussed the positive effect of volatility in systems . At the time I was not aware of the notion Heart Rate Variability (HRV) but had noticed the fact that a healthy person must have a faster recovery to baseline after strenuous exercise, which necessarily implies that an increased variance of the heart rate would be better, not worse, at least up to a point . This quality is shared in all systems, even revenues of companies .

I' ve seen quite a bit of discussions of HRV on the web but could not quite understand them . Why? First, as we can see below, HRV is way too noisy to be informative!





HRV according to the Apple Watch : It swings more than one order of magnitude in half a day!
We use their definition of STD of interarrival time .

First, HRV can be intuitively the standard deviation of the number of beats in a narrow interval [t, t+Δt], as done with a Poisson process. However it is defined here as a function of $\Delta\tau$, variations in arrival time $\tau_1, \tau_2, \ldots, \tau_n$.
Whether the metric is scaled is here irrelevant.

$$\text{HRV} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\tau_i - \tau_{i+1})^2 - \left(\frac{1}{n}\sum_{i=1}^{n}\tau_i - \tau_{i+1}\right)^2}$$

Data

In[∘]:=
```
Clear[data];
data = Import[
  "/Users/nntaleb/Downloads/HealthAutoExport-2020-10-23-2021-10-16 Data.csv"];
```

We correct by removing long blocks of empty time

In[288]:=
```
time = Drop[Transpose[data][[1]], 1];
dtunadjusted = Drop[Transpose[data][[1]], 1] // Differences;
HRVunadjusted = Drop[Transpose[data][[2]], 2];
res = Transpose[{dtunadjusted, HRVunadjusted}];
Xdt = Select[res, #[[1]] < .1 &];
dt = Transpose[Xdt][[1]];
HRV = Transpose[Xdt][[2]];
dtHRV = Transpose[{dt, beats}];
```

Check on the Existence of Moments

In[308]:=
```
Max[HRV] / Min[HRV]
```

Out[308]=
```
40.9587
```

A bit excessive $\frac{\text{Max}}{\text{Min}}$! Let us verify that it is NOT a power law.

An interesting graphical tool, which we can use to have an idea of the behavior of moments in a given sample, is the so-called Maximum-to-Sum plot, or MS Plot. The MS Plot relies on simple consequence of the law of large numbers [11]. For a sequence $X_1, X_2, ..., X_n$ of nonnegative i.i.d. random variables, if for $p = 1, 2, 3...$,

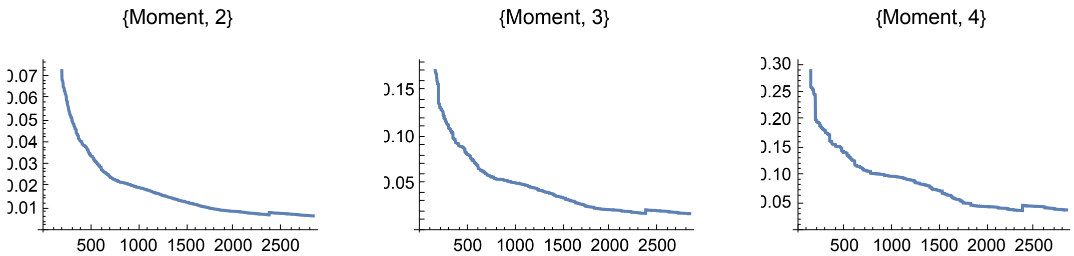
---

$E[X^p] < \infty$, then $R_n^p = M_n^p / S_n^p \to^{a.s.} 0$ as $n \to \infty$, where $S_n^p = \sum_{i=1}^n X_i^p$ and $M_n^p = \max(X_1^p, ..., X_n^p)$.

In[305]:=
```
MP[X_, p_, n_] := Max[Table[Abs[X[[i]]]^p, {i, 1, n}]] / Sum[(Abs[X[[i]]]^p), {i, 1, n}]
```

In[313]:=
```
GraphicsRow[Table[ListLinePlot[
   Table[MP[HRV, i, n], {n, 1, Length[HRV]}], PlotLabel → {Moment, i}], {i, 2, 4}]]
```

Out[313]=



We do not see power laws! Great news in spite of behavior. It must be Lognormal. Why?
Because the Lognormal is the only distribution that has nasty behaviors while having ALL moments!

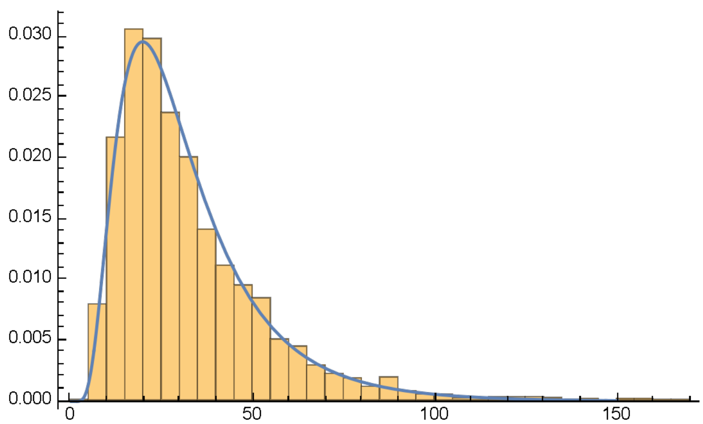
Lognormal Fitting

In[299]:=
```
emp = EmpiricalDistribution[HRV]
dist = LogNormalDistribution[μ, σ] /.
  Solve[{Mean[HRV] == e^(μ + σ²/2), Variance[HRV] == e^(2 μ + σ²) (-1 + e^(σ²))},
    {μ, σ}, Assumptions → {μ > 0, 1 > σ > 0}][[1]]
Show[ Histogram[HRV, Automatic, "PDF", PlotRange → All],
 Plot[PDF[dist, x], {x, 0, 150}, PlotRange → All]]
```

Out[299]= DataDistribution[Type: Empirical Data points: 2829]

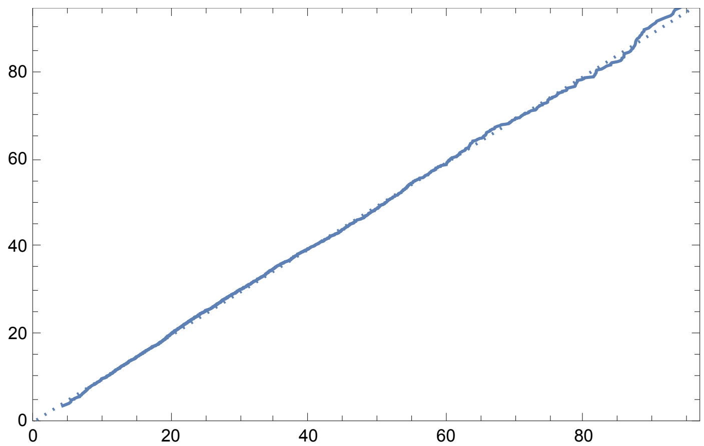
Out[300]= LogNormalDistribution[3.31242, 0.581652]

Out[301]=



In[304]:=
```
QuantilePlot[ dist, HRV]
```

Out[304]=



What does it mean?

If the data is Lognormal, it is improper to use regression on non-logged variables. Since data is impeccably Lognormal, let us work with transforms.

In[316]:=
```
{Log[HRV] // Skewness, Log[HRV] // Kurtosis, Max[Log[HRV]] / Min[Log[HRV]],}
```

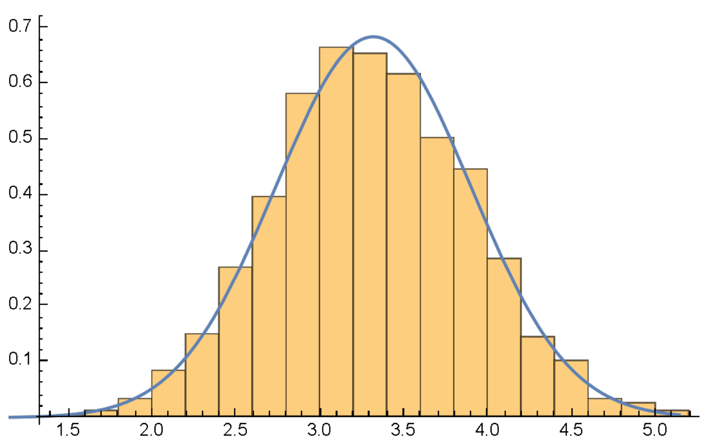
Out[316]=
```
{0.0749801, 2.84408, 3.62214, Null}
```

In[338]:=
```
emp2 = EmpiricalDistribution[Log[HRV]]
distlog = NormalDistribution[μ, σ] /. μ → dist⟦1⟧ /. σ → dist⟦2⟧
Show[ Histogram[Log[HRV], Automatic, "PDF", PlotRange → All],
 Plot[PDF[distlog, x], {x, 0, Log[170]}, PlotRange → All]]
```

Out[338]=
```
DataDistribution[ Type: Empirical  Data points: 2829 ]
```

Out[339]=
```
NormalDistribution[3.31242, 0.581652]
```

Out[340]=



Works!