Notes on volatility targeting in rsystrade #rsystrade #volatility

This document lays out how volatility targeting is done in **rsystrade**. This method takes its outset in the system presented in Robert Carver's book *Systematic Trading*.

It then proceeds to drill deeper into certain theoretical aspects, to help understand the relations in mathematical terms. Some of the assumptions underlying Carver's system are examined.

> **Note**
>
> In **rsystrade** we say *signal* when Carver might say *forecast*.
> Consequently we say *Signal Diversification Multiplier* instead of *Forecast Diversification Multiplier*.
> We say *clamping* where Carver migh say *capping/flooring*.

Volatility targeting in rsystrade

Signals undergo four stages of scaling, in this order:

1. Individual rules may, if they output a signal in *price units*, be volatility adjusted by dividing the signal by the *standard deviation in price units of recent returns* of the instrument. This scaling is baked into the rule, which takes volatility in price units as input. The idea is to get a signal which is proportional to the Sharpe Ratio. The signal will be weaker the more volatility.
   - Standard deviations are calculated for a rolling window of e.g. 25 days.
   - As an alternative measure of volatility we could use the *Exponentially Weighted Movin Average of the squared returns*. Here a window length of 36 days would be equivalent of a 25 days window for standard deviations.
2. The (volatility adjusted) signal is scaled by a *signal normalization factor*. This is calculated by dividing
   - the *target expected absolute signal value* (1 for rsystrade, 10 for pysystemtrade)
   - by the rolling (or expanding) mean of the median of a cross section of signals for the individual rule applied to a pool of instruments.
   - This normalized signal is then clamped between -2 and 2.
3. The weighted mean of all normalized and clamped signals for a single instrument is scaled by a *signal diversification multiplier* based on weighted correlations of signals for that instrument.
4. Finally, the combined signal is multiplied by a volatility scalar, which is the target risk divided by the instrument risk.
   - This signal is also clamped between -2 and 2.

Both the normalized individual signals and the scaled combined signal should be clamped. The default limits are -2 below and 2 above.

> **Note**
>
> This is the outset for the framework. But we want to be able to test different variations for each stage of scaling, i.e. different measures of dispersion.
> Possible alternative:
>
> - Calculate *instrument target risk* from *system target risk*.
> - Calculate *required leverage factor* for an instrument as *instrument risk target* divided my *instrument risk*.

Signals from individual rules

This section describes how to scale the signal for each rule.

A *signal* is the raw volatility adjusted signal multiplied by a *signal normalization factor*.

1. Volatility adjusted signal:
    - Divide signal value at time t by standard deviation of recent window of returns.
    - We scale the signal by the price volatility to reflect that we want position sizes that give us the over all amount of risk we want. So we want bigger positions in a less risky instruments, and smaller positions in more risky instruments.
2. Calculate *signal normalization factor*:
    - Take median of pooled absolute signals at time t.
    - Calculate mean of a recent window of medians.
    - To get normalizing factor: Divide target expected signal value (default 1 for rsystrade) by the mean of recent window of medians of pooled signals.
    - The *signal normalization factor* is calibrated for each rule, e.g. once a year.
        - Calibrate to expanding window.
        - Alternative: Rolling window.
        - Minimum 1 year of data recommended.
        - Or we might recalibrate the *normalization factor* based on other metrics, such as the detection of a new regime.
    - To avoid a jump in signals when *signal normalization factor* is recalculated, smoothing can be applied to the signal.
        - We could also recalculate the *signal normalization factor* daily instead, but
            - consider computation time for pooled estimate.
            - this might be overfitting.

The normalized signal should be limited above and below. In **rsystrade** this is called *clamping*. The default limits are -2 below and 2 above.

So far we have a signal with an expected value of 1, a min of -2 and a max of 2.

Pooling instruments

- Normalizing for each rule might be calibrated on a number of instruments.
    - The idea is to scale the expected signal value to 1.

        - Take median of absolute signals of a cross section across instruments.
        - Take mean of meadians on a rolling basis (moving average).
        - Devide signal by mean of medians and multiply by target.
    - Expected within which domain?
        - The domain should be general enough, but not too general, and specific enough, but not too specific...!
        - We don't know enough about how a single instrument behaves "in general", so scalars based on single instruments are too genera.
        - Should the scalar be based on as many instruments as possible? Is that too general? E.g., what if an asset is clearly cyclical, but an average of all assets we can imagine is not cyclical?
- Which instruments should be included when calculating the scalars?
- Different scalars for different asset classes?
- Other groupings for scalars?
- When should scalars be updated?
    - On a rolling (moving average) basis, at fixed time intervals (i.e. once a year, or daily)?
    - When some regime change is detected?

Diversification Multiplier

The rest of this document deals with the question of combining signals, when multiple rules are applied to the same instrument.

- The combined signal for each instrument is calculated as a weighted average of all the signals for that instrument.
- The weighted average is then multiplied by a *Signal Diversification Multiplier*.
- While the individual signals are volatility adjusted to the *expected standard deviation of daily instrument percentage returns* (aka *price volatility*), the combined signal is adjusted to the correlations of signals.
- Like the individual signals, the combined signal is clamped between -2 and 2.

Why not simply divide target of combined signal by dispersion of combined signal to get a *Required Leverage Target* instead of using the *Signal Diversification Multiplier*?

- Because:
    - We are calculating the *Required Leverage Factor* for each individual rule by pooling instruments using <u>*Median*</u> *Absolute Value* for cross sections and then taking the <u>mean</u> of pooled medians. By simply dividing the target of combined signal by dispersion of the combined signal instead of using the *Signal Diversification Multiplier* we would get a RLF based on a specific instrument, and we want something more general.
    - We don't want to calculate the "exact" multiplier for each individual instrument at any given point in time. We want to calculate the *expected* multiplier across pools of instruments as well as across time periods. If we were to calculate multipliers for different individual instruments, they would (should) be distributed around the expected value.
        - Yet, we may want to adjust to different regimes over time.

- We may want to identify different classes of instruments which share the same expected multiplier.

Still, why use the correlation matrix rather than the variance–covariance matrix?

> **Note**
>
> - **pysystemtrade** doesn't update the FDM (SDM) all the time, only once a year.
> - The normalization factor (*Required Leverage Factor*) should then be smoothed (using shrinkeage)

The purpose of this note is to investigate the exact practical and theoretical difference between the two methods.

Notation

- X_i: A stochastic variable.
- $\mathbf{x}_i$: An $(n \times 1)$ signal vector produced by X_i.
- $x_{i,t}$: A signal value produced by X_i at time t.
- $\mathbf{x}_t$: A $(p \times 1)$ vector of signal values produced by $\{X_i\}_i^p$ at time t.
- $\mathbf{X} := [X_1, X_2, \ldots, X_p]^T$: A stochastic vector.
- $\mathbf{w} := [w_1, w_2, \ldots, w_p]^T$: A $(p \times 1)$ vector of weights summing to 1.
- $\tilde{X}_i := w_i X_i$: A weighted stochastic variable.
- $\tilde{\mathbf{X}}$: A $(p \times 1)$ weighted stochastic vector:
  - $$\tilde{\mathbf{X}} := \mathbf{w} \circ \mathbf{X} \equiv [w_1 X_1, w_2 X_2, \ldots, w_p X_p]^T$$
  - > **Note**
    >
    > - Each element in $\mathbf{X}$ is a stochastic vector, so each element in $\tilde{\mathbf{X}}$ is a weighted stochastic vector.
    > - $\mathbf{wX}$ is a weighted average of the elements in $\mathbf{X}$.
    > - Therefore, $\sum_i^p \tilde{\mathbf{X}}_i$ is a weighted average of the elements in $\mathbf{X}$.
    > - We don't divide by p, because the weights sum to 1.
    > - $\sum_i^p \tilde{\mathbf{x}}_{t,i}$ is a weighted average of the signal values of all the p signals at time t.
- $\tilde{\mathbf{x}}_i$: An $(n \times 1)$ vector of weighted signals produced by $\tilde{\mathbf{X}}_i$.
- $\tilde{\mathbf{x}}_{i,t}$: A signal value produced by $\tilde{X}_i$ at time t.
- $\tilde{\mathbf{x}}_t$: A $(p \times 1)$ vector of signal values produced by $\{\tilde{\mathbf{X}}_i\}_i^p$ at time t.
- $\hat{X}'$: Scaled, combined, weighted signal. Stochastic variable.
- MAV: Mean Absolute Value.
  - $$\mathrm{MAV}[X_i] := \frac{1}{n} \sum_t^n |\mathbf{x}_{i,t}|$$
  -

$$\mathrm{MAV}[\tilde{\mathbf{X}}_t] := \frac{1}{p}\sum_j^p |\tilde{X}_{t,j}|$$

- MAV_τ: Target MAV.
- MAD: Median Absolute Deviation.
- AAD: Average Absolute Deviation, aka Mean Absolute Deviation (MAD).
- $X_i^{'}$: Scaled X_i.
  - $$X_i^{'} := X_i \frac{\mathrm{MAV}_\tau}{\mathrm{MAV}[X_i]}$$
- $\hat{X} := \sum_i^p \tilde{\mathbf{X}}_i$: Combined, weighted signal. Stochastic variable.
- $\mathbf{\Sigma}$: Variance-Covariance matrix of $(X_1, X_2, \ldots, X_p)$:
  - $$\mathbf{\Sigma} := \mathrm{cov}(\mathbf{X})$$
- $D := \sqrt{\mathrm{diag}(\Sigma)}$. Diagonal matrix with standard deviations in diagonal.
- $H := \mathrm{cor}(\mathbf{X})$
- K: A constant $\{K \mid \mathrm{MAV} = \sigma K\}$
  - The expected absolute value of a normal random variable with mean $\mu = 0$ and standard deviation σ is $\sigma\sqrt{\frac{2}{\pi}}$:
    - For $X_i \sim N(0, \sigma)$, $K_{X_i} = \sqrt{\frac{2}{\pi}}$

Scaled, combined signal at time t. This is what we are after!

1. $$\hat{X}^{'} := \left(\sum_i^p \tilde{X}_i\right)\frac{\mathrm{MAV}_\tau}{\mathrm{MAV}[\tilde{\mathbf{X}}]}$$
2. $$= \left(\sum_i^p \tilde{X}_i\right)\frac{\mathrm{MAV}_\tau}{\sigma_{\tilde{X}} \cdot K}$$
3. $$\equiv \left(\sum_i^p \tilde{X}_i\right)\frac{\mathrm{MAV}_\tau}{\sqrt{\mathbf{w}^T\mathbf{\Sigma}\mathbf{w}} \cdot K}$$
4. $$= \left(\sum_i^p \tilde{X}_i\right)\frac{\mathrm{MAV}_\tau}{\sqrt{\mathbf{w}^T\mathbf{DHDw}} \cdot K}$$

Here we are using
$u := (w \circ s)$
$s := \sqrt{\mathrm{diag}(\mathbf{\Sigma})}$

$$uHu \equiv w\mathbf{\Sigma}w$$

where $w \circ s$ is the element-wise multiplication of the vector of weights and the vector of standard deviations, and

$$w\mathbf{\Sigma}w \equiv V[\tilde{\mathbf{X}}] \equiv \sigma^2_{\tilde{X}}$$

Notice, that

- Equality 1 defines $\hat{X}'$.
- Equality 2 is true by the definition of K. K is defined as the value that makes the equality true.
- Equality 3 is an identity by the definition of the variance of a weighted sum.
- Equality 4 is true by the relation between a variance-covariance matrix and a correlation matrix.

Notice, that the practiced method is

$$\text{RaCF} \cdot \text{MAV}_\tau \cdot \text{SDM} \equiv \left(\sum_i^p \tilde{X}_i \right) \cdot \text{MAV}_\tau \cdot \frac{1}{\sqrt{\mathbf{w}^T \mathbf{H} \mathbf{w}}}$$

where

- $\text{RaCF} := \left(\sum_i^p \tilde{X}_i \right)$ (Raw Combined Forecast).
- $\text{SDM} := \frac{1}{\sqrt{\mathbf{w}^T \mathbf{H} \mathbf{w}}}$.
  So the difference between the theoretical method and the practiced method is determined by the contribution of $\mathbf{D}$ versus the contribution of K. See Comparisons below.

Test[1]:

Check that $\sigma_{\tilde{X}} \cdot K = \text{MAV}[\tilde{X}]$.
When $\text{mean}(\tilde{X}) = 0$, K should be $2/\pi$.
Here, the mean of $\tilde{X}$ is not zero, so we shift $\tilde{X}$ to get a zero mean.
Now the expected difference should be zero.
0.01167052 is close to zero. Close enough?

```
## Let V be the variance-covariance matrix (\{Sigma}) of the signal.

## 1)
K <- sqrt(2/pi)
## Shift to zero mean
comb_sig_demean <- comb_signals - mean(comb_signals)
## The expected difference is zero
(sd(comb_sig_demean) * K) - mean(abs(comb_sig_demean))
```

```
#0.01167052
```

These should be identical:

```
## 2a)
sd(comb_signals)

## 2b)
sqrt(var(comb_signals))[1,1]

## 3)
sqrt(
```

```
  crossprod(
    t(w %*% V),
    W
  )
)[1,1]

## 4)
ws <- w * (diag(sqrt(V))) ## weighted st. devs
sqrt(
  crossprod(
    t(ws %*% H),
    ws
  )
)[1,1]
```

```
#0.7600743
#0.7600743
#0.7600743
#0.7600743
```

Comment

- Let's say we are looking at p rules applied to a single instrument, and we want to calculate the combined signal.
- Above, each $\mathbf{X}_i$ could represent a signal process for a single rule applied to the instrument.
- Then $\tilde{\mathbf{X}}_i$ would be the weighted signal for that rule.
- $\hat{X}'$ is the scaled sum of the weighted signals.
- So when we calculate $\hat{X}'$ above, the *signal normalization factor* is based on $\mathrm{MAV}[\tilde{\mathbf{X}}]$.
- How is $\mathrm{MAV}[\tilde{\mathbf{X}}]$ calculated?
- The correlation matrix we use in practice is a sort of volatility standardized variance-covariance matrix. [?]

  > **Todo**
  >
  > Fix this. Train of thought not completed...

Comparisons

Comparison 1: Compare standard deviation and MAV

Signal follows a normal distribution

- To study the difference between *standard deviation* and MAV (equality 2 above):
  - For different distributions of $X_i \in \{X_i\}_{i=1}^p$, compute

$$K = \frac{\mathrm{MAV}[\tilde{\mathbf{X}}]}{\sigma_{\tilde{X}}}$$

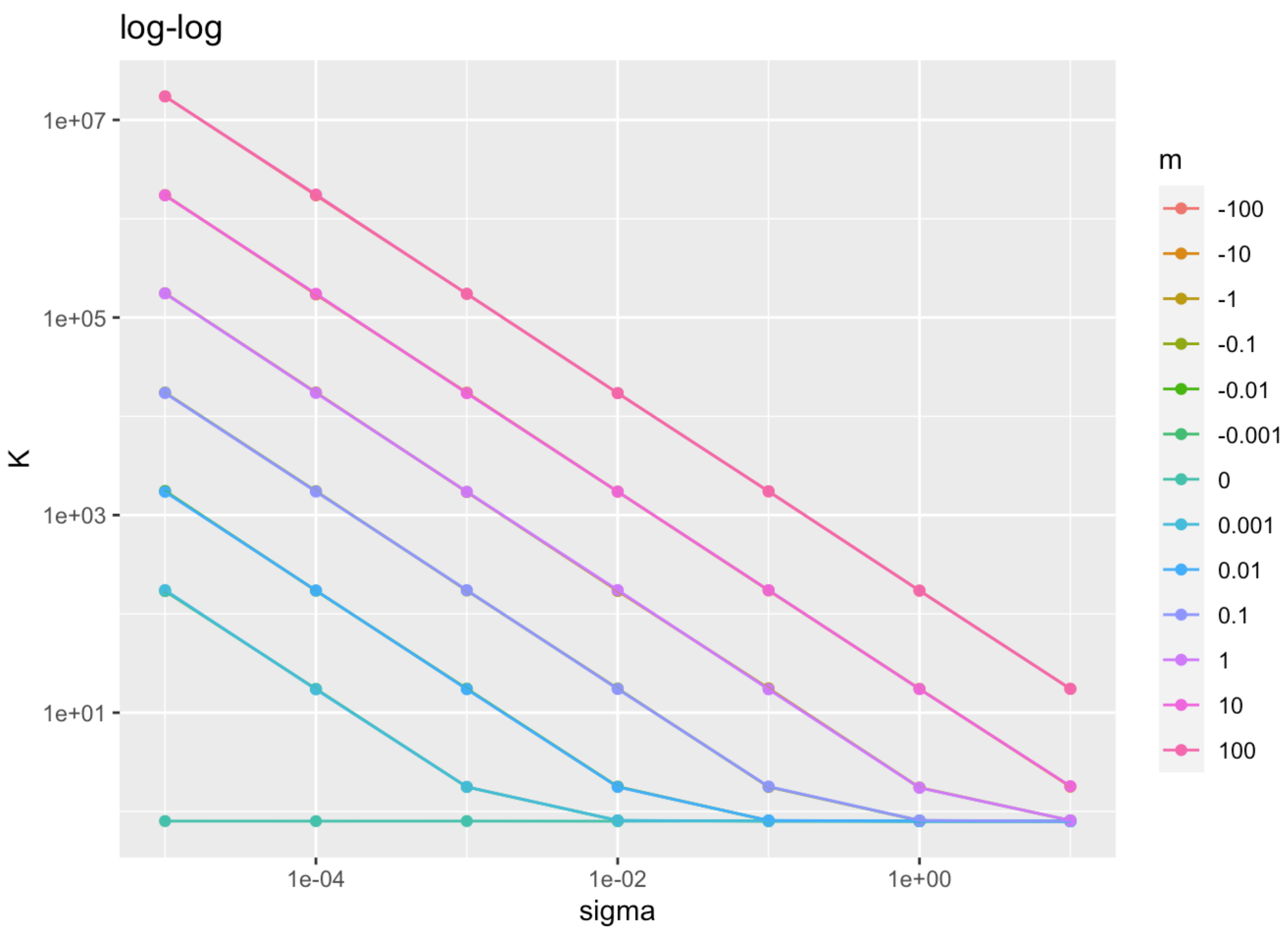
    - wrt. μ

- wrt σ
- Assume all individual signals are i.i.d.

- **Note**

  Here we are looking at the combined signal.
  Comparing *standard deviation* and MAV is also relevant to the volatility targeting of each individual signal.
  However, here we are not just comparing different metrics of dispersion, but also comparing dispersion of signal in one case and returns in the other. Signals are often in price units, however.
  Notice that we are volatility adjusting each signal by dividing the signal by the standard deviation of a recent window of returns. But we are multiplying by a *signal normalization factor* based on MAV of a pool of signals.

  - Why are we not adjusting the signal by MAV instead of sd?
  - Compare
    - standard deviation of a recent window of returns
    - MAV of a pool of signals

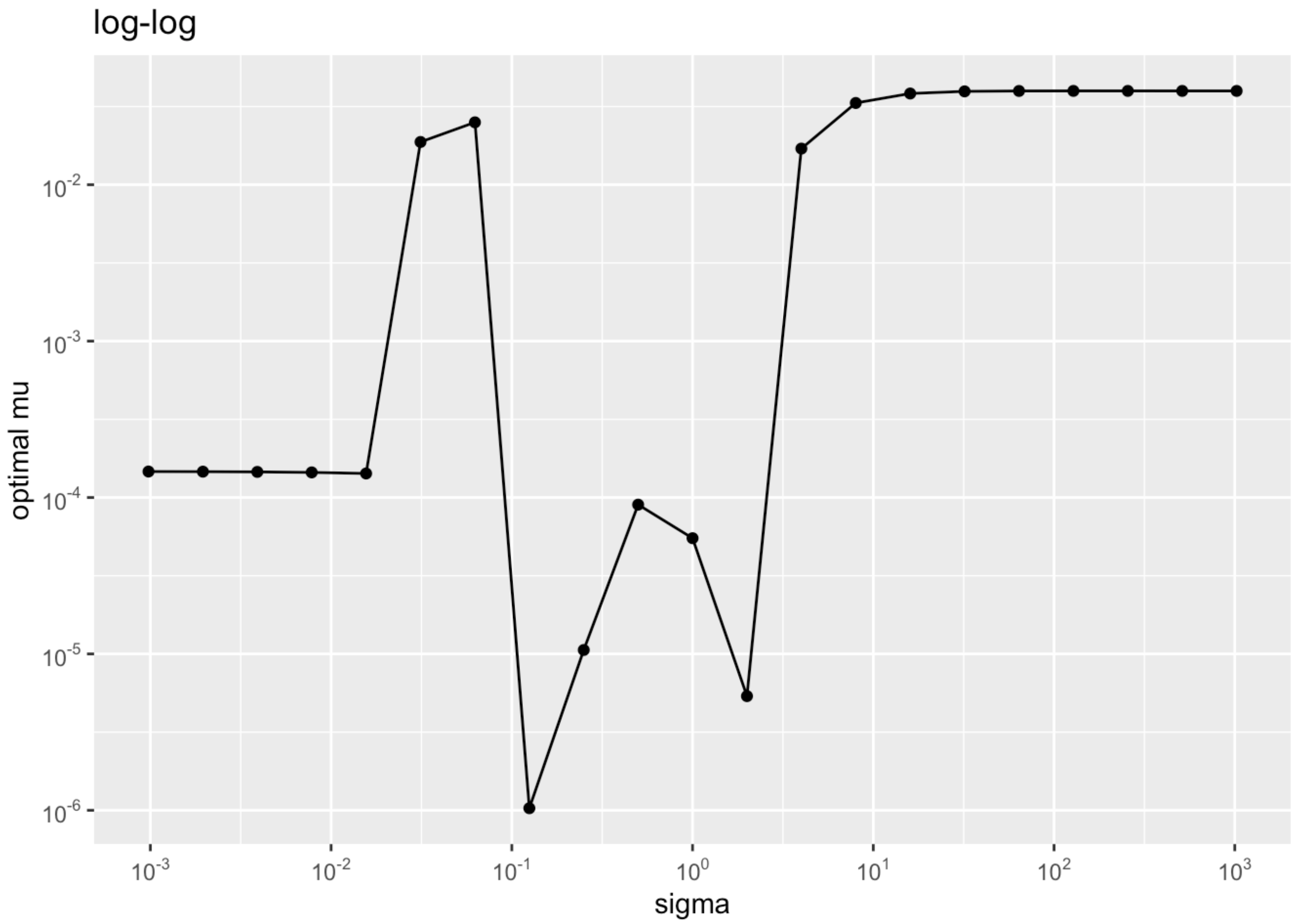


For now it looks like:

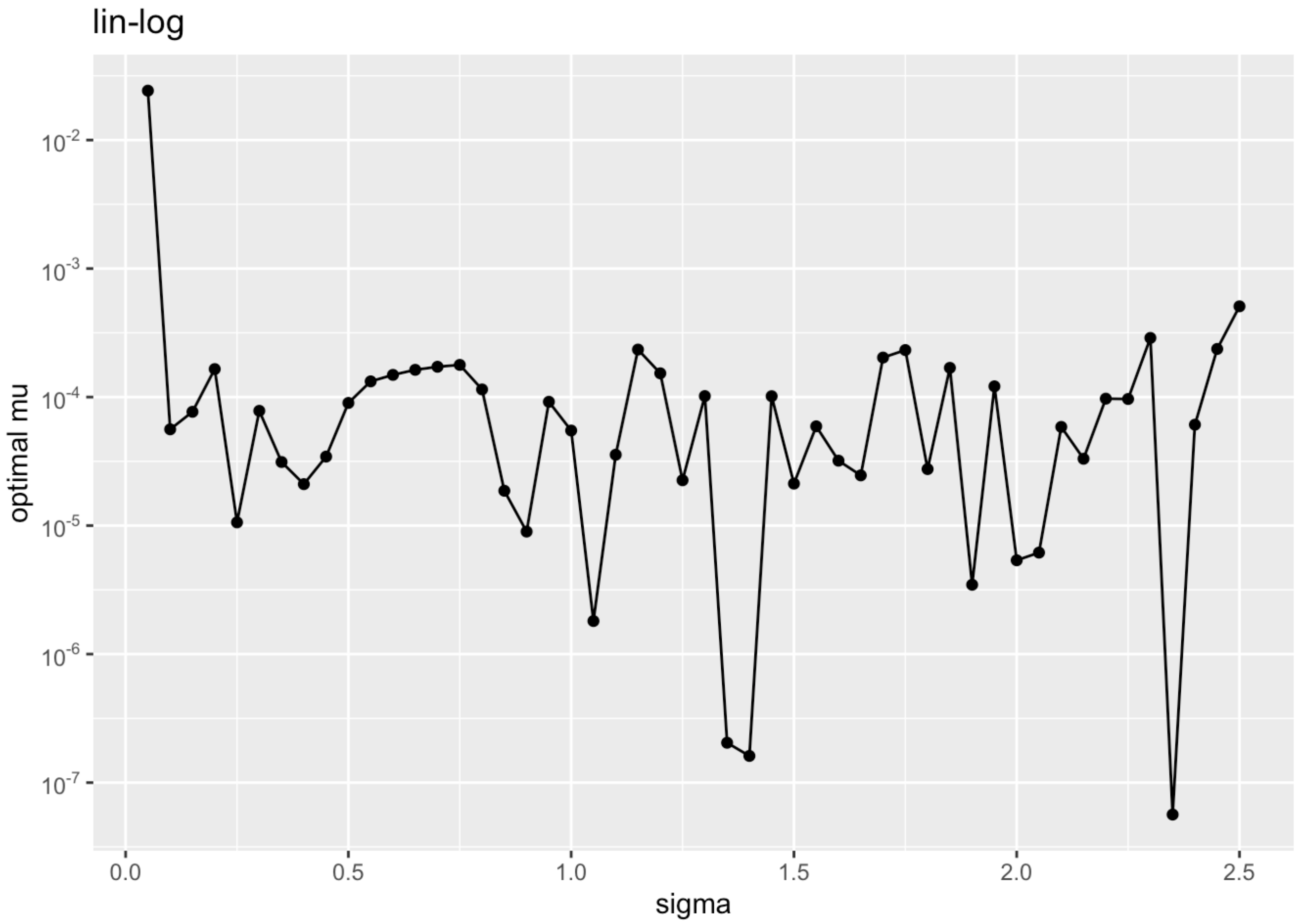
- K is close to 1 if (e.g):
  - $\sigma > 0.01$ and $0 < m < 0.001$.

    - $\sigma > 1$ and $0 < \text{abs}(m) < 0.1$.
- K close to 1 means that MAV and sd are equivalent.

For which μ is $K = 1$?



Inspect the mid range:

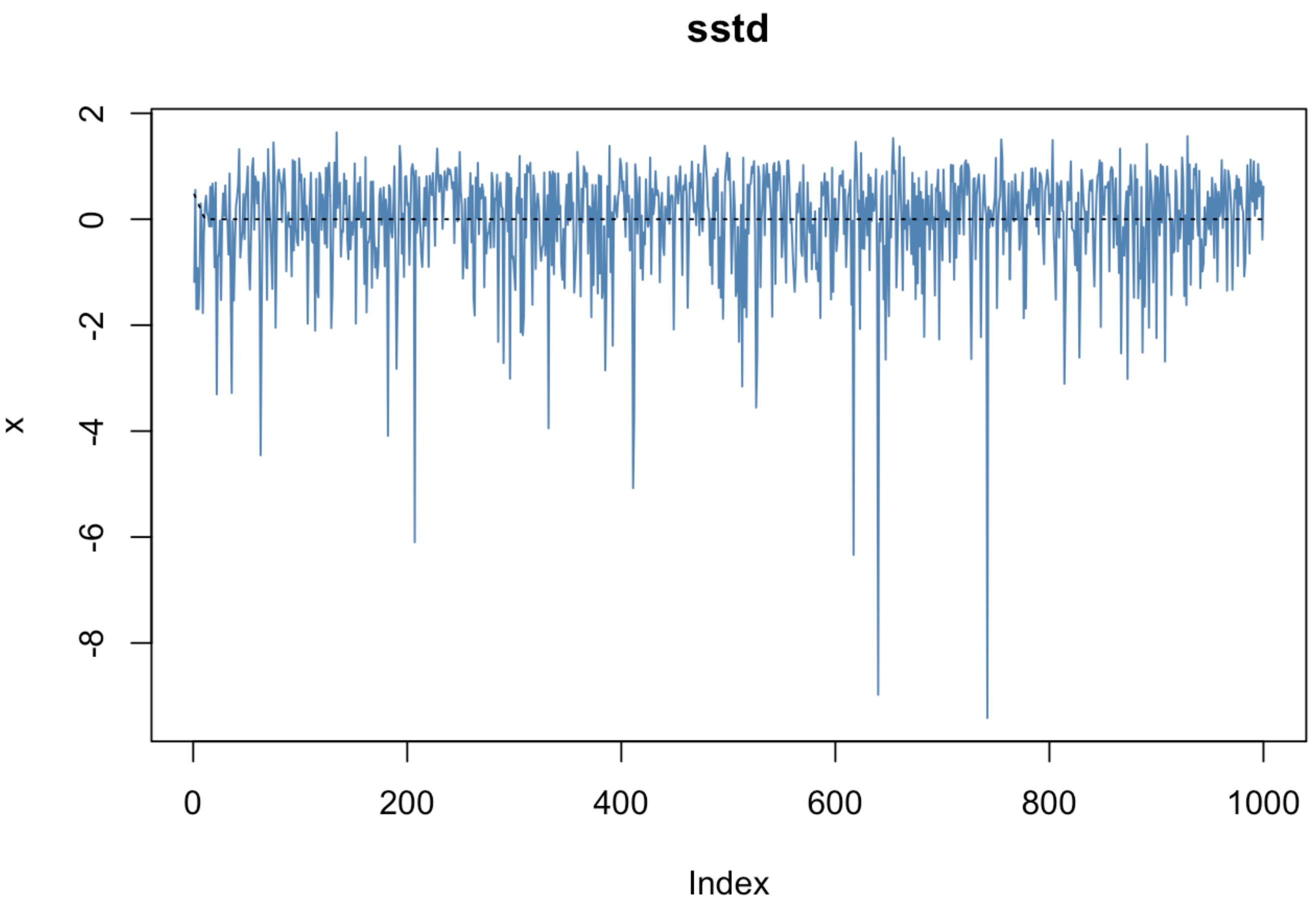


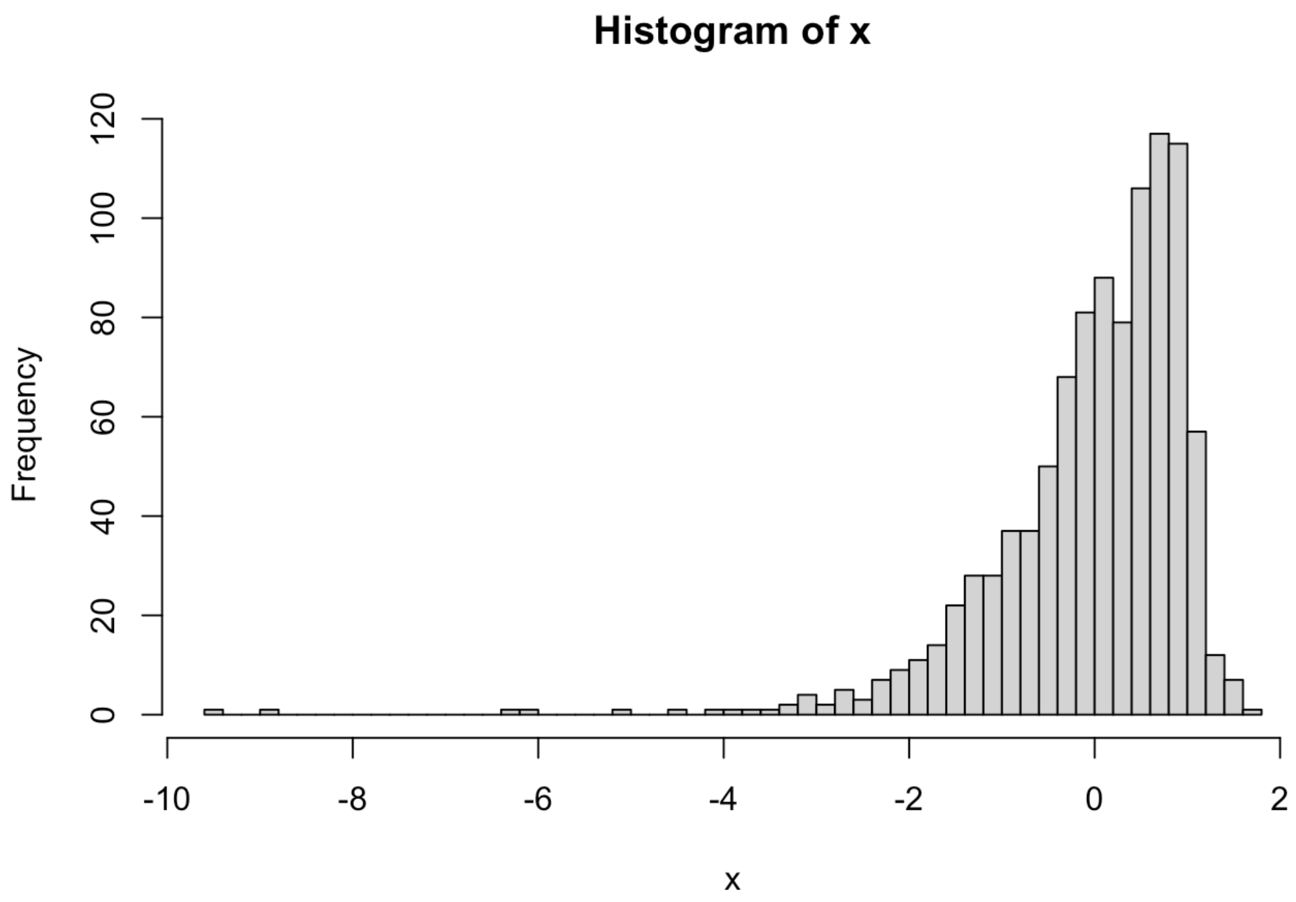
Conclusion

- K is close to 1 if (e.g):
  - $\sigma > 0.01$ and $0 < m < 0.001$.
  - $\sigma > 1$ and $0 < \text{abs}(m) < 0.1$.
- K close to 1 means that MAV and sd are equivalent.
- WARNING: K can be huge, if the mean of the individual signals is different from 0, and the standard deviation is close to 0.
- The optimal μ is quite unstable wrt. σ for different draws of X.
  - Especially in the range $\sigma \in [0.01, 4]$.
    - This means that anywhere near a standard normal distribution, K is unstable.
- If σ is large (≥ 10), then optimal μ seems stable, and therefore K should also be stable wrt. σ.
- For $\sigma \geq 10$ the optimal μ seems to be around 0.04.

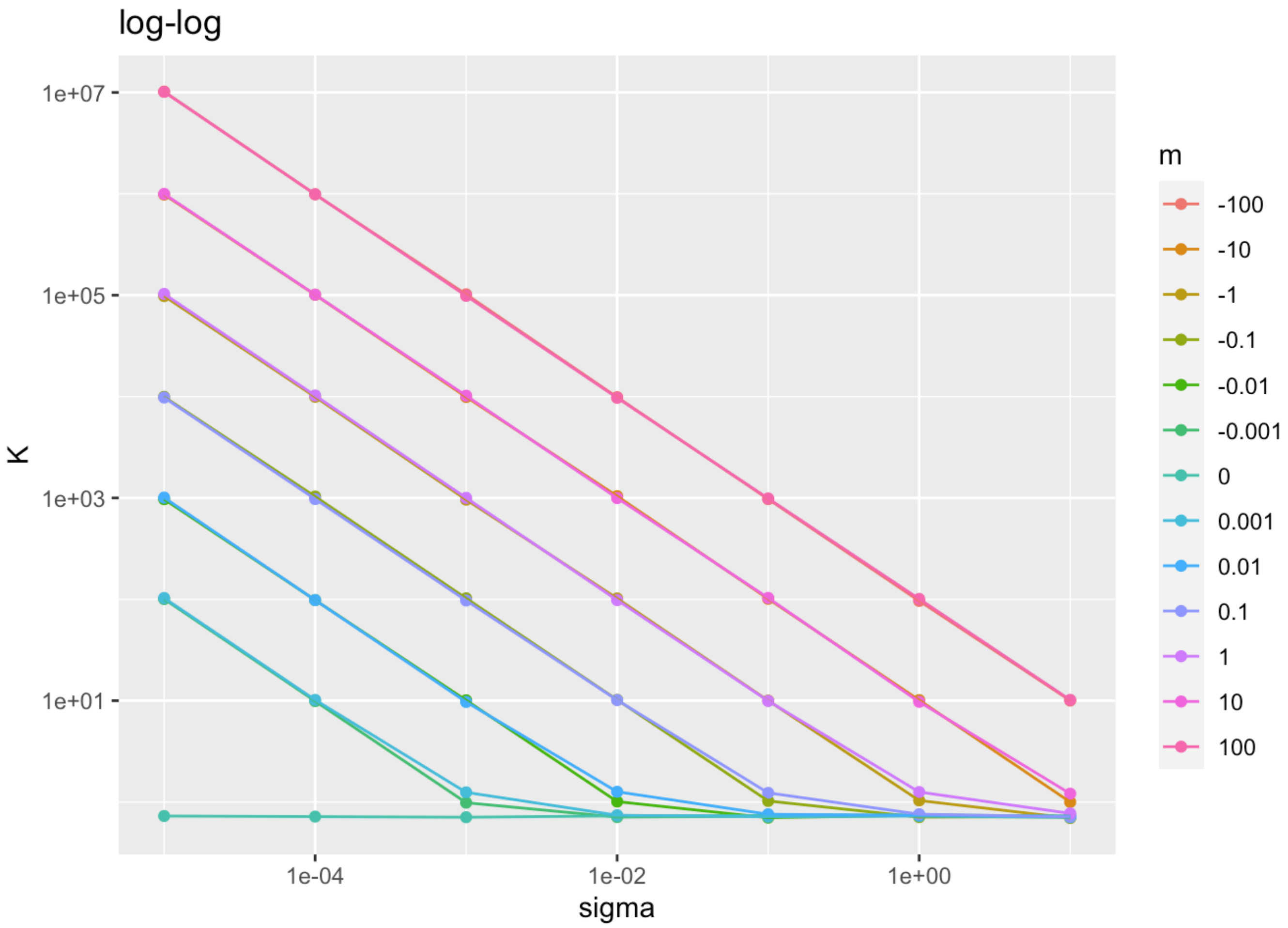
Signal follows a skewed t-distribution

```
p = 3
n = 10000
nu = 5
xi = 0.4
```


sstd
x
Index
2
0
-2
-4
-6
-8
0
200
400
600
800
1000


Histogram of x
Frequency
x
0
20
40
60
80
100
120
-10
-8
-6
-4
-2
0
2

```
n = 10000
```



Conclusion

We see a similar pattern as with normal distributed signals, with the difference that K differs a bit for opposite signed values of μ.

By eye it looks like K actually tends to be bigger for normal distributed signals than for heavily skewed signals.

Comparison 2: $\mathbf{D}$ vs K

- Study to which degree the contributions of $\mathbf{D}$ and K cancel each other out.
  - $$\frac{\sqrt{\mathbf{w}^T\mathbf{DHDw}} \cdot K}{\sqrt{\mathbf{w}^T\mathbf{Hw}}} = 1?$$
  - If the contributions of $\mathbf{D}$ and K cancel each other out, we have:

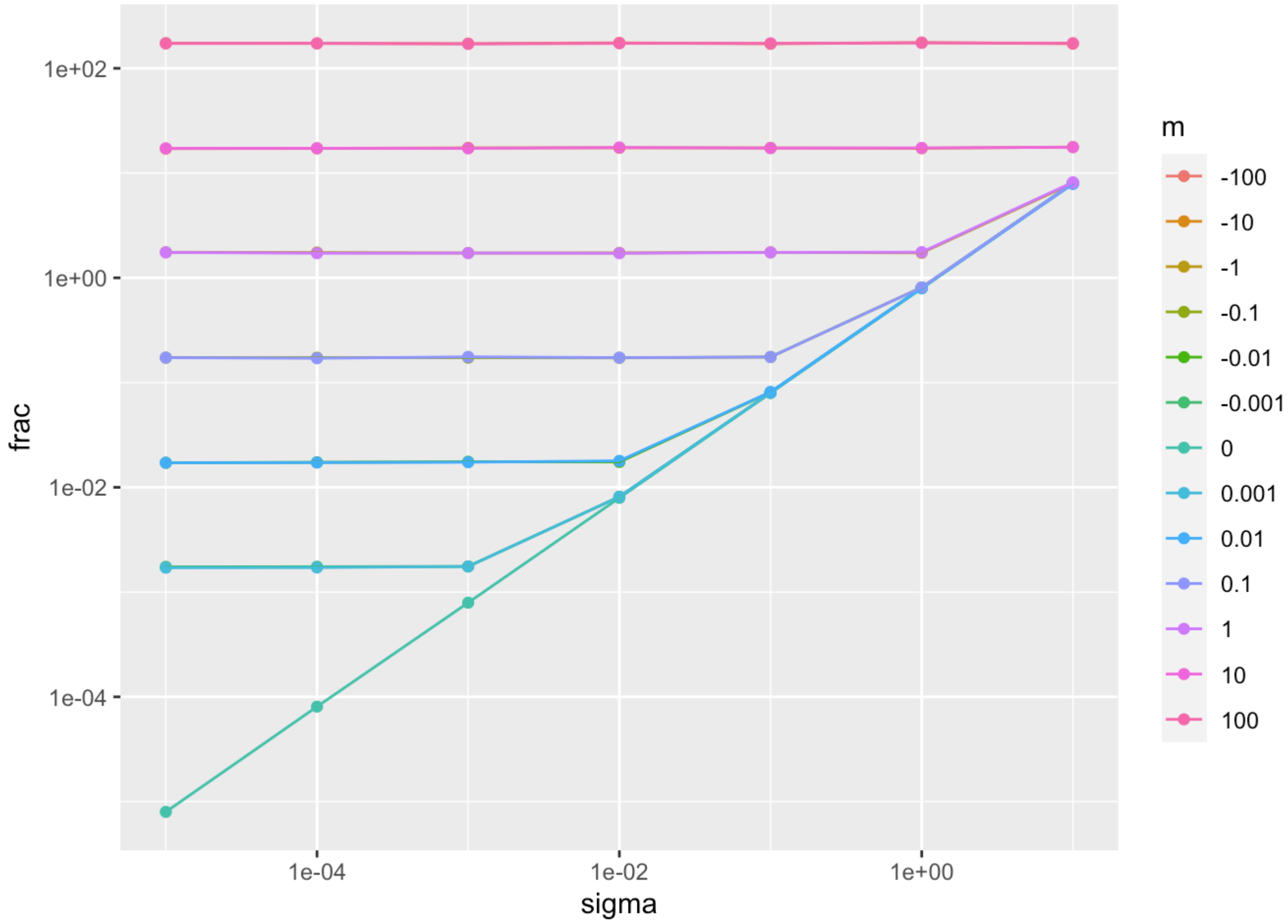
$$= \left(\sum_i^p \tilde{X}_i\right) \frac{\text{MAV}_\tau}{\sqrt{\mathbf{w}^T\mathbf{\Sigma w}} \cdot K}$$

$$= \left(\sum_i^p \tilde{X}_i\right) \frac{\text{MAV}_\tau}{\sqrt{\mathbf{w}^T\mathbf{DHDw}} \cdot K}$$

$$= \left(\sum_i^p \tilde{X}_i\right) \frac{\text{MAV}_\tau}{\sqrt{\mathbf{w}^T\mathbf{Hw}}}$$

- ...which would mean, that the two methods are indeed identical.

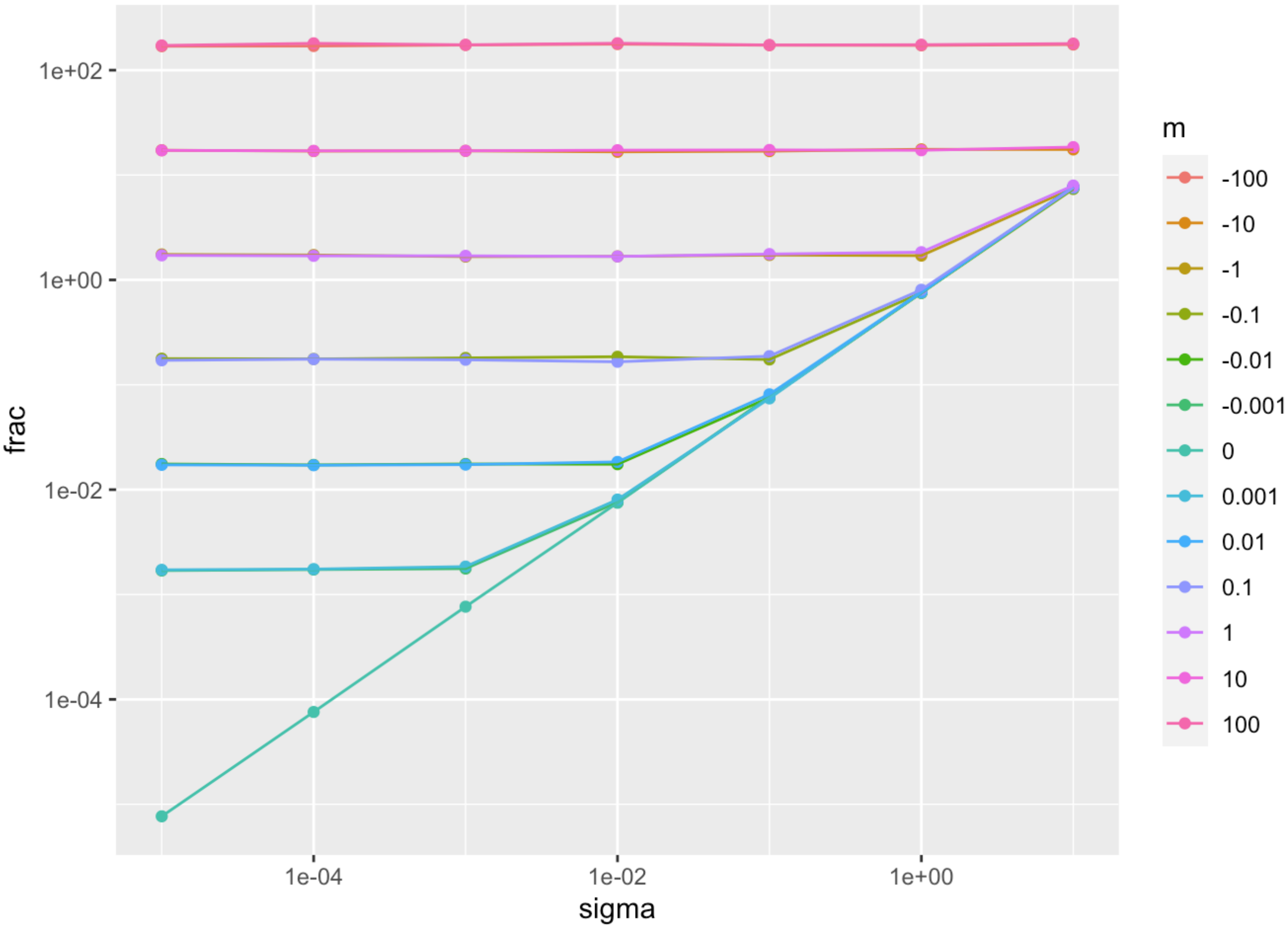
Signal follows a normal distribution



Conclusion

- The conclusion for normal distributed signals is analogous to the one in comparison 1 above.
- The contributions of $\mathbf{D}$ and K cancel each other out, if (e.g.):
  - $\sigma > 0.01$ and $0m < 0.001$.
  - $\sigma > 1$ and $\mathrm{abs}(m) < 0.1$.
- WARNING: The difference can be huge, if the mean of the individual signals is different from 0, and the standard deviation is close to 0.
- If we use $w^T H w$ instead of $w^T \Sigma w$, the diversification multiplier can be many (millions of) orders of magnitude bigger than the theoretical MAV that we are targeting, of σ is small and μ is not 0.
- IMPORTANT: For these reasons, the target mean average signal should be bigger than 1. A target of 10 (as in pysystemtrade) seems good, or even 100, which may also be more intuitive (200 means double, as in 200 percent).

Signal follows a skewed t-distribution

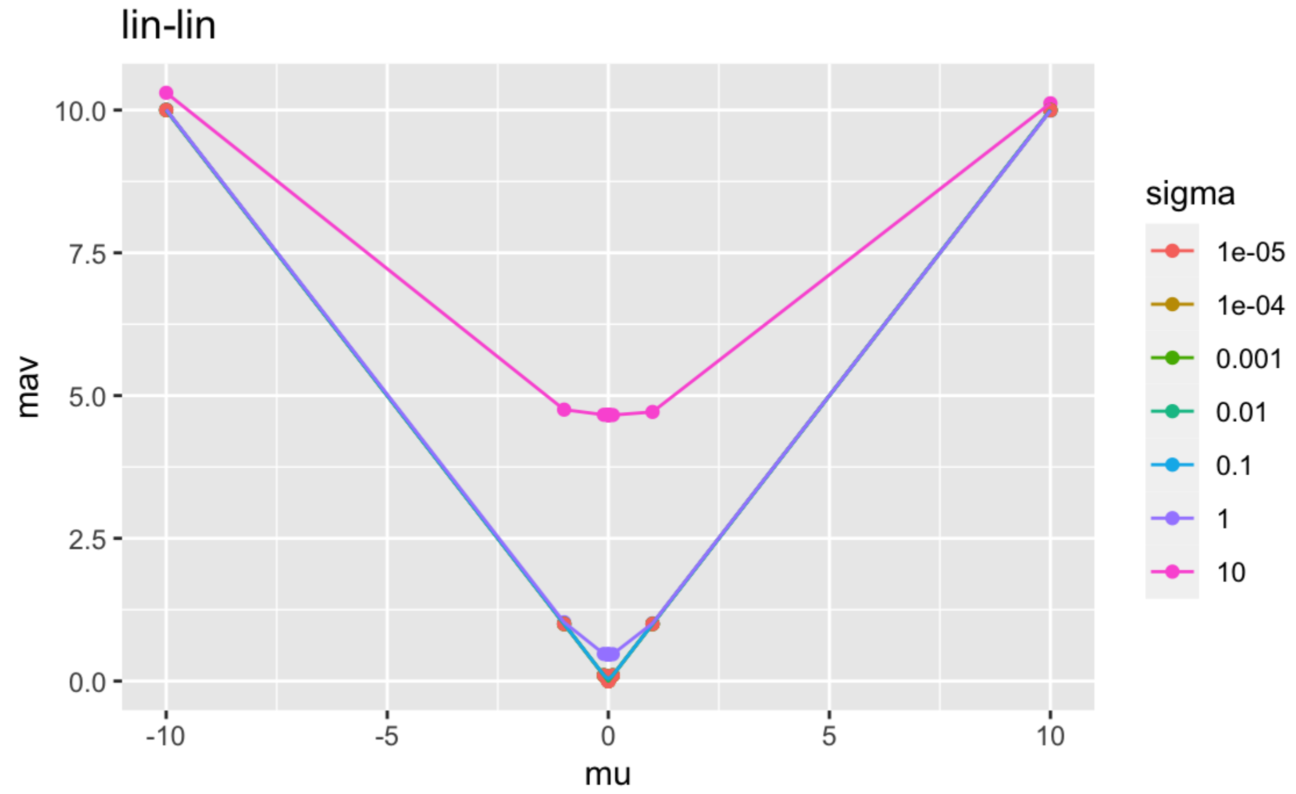


Conclusion

- For the heavily skewed signals, the picture is quite different than for normal distributed data.
- For $\mu = 0$, the fraction is approximately equal to σ.
- The further μ is from 0, the bigger the fraction.
- From visual inspection: The fraction is quite stable around 1 when μ is somewhere between 0.1 and 1, and $\sigma < 1$.

Comparison 3: MAV wrt. μ

Signal follows a normal distribution



Conclusion

The bigger the sd,

- the bigger `mean(abs(signal)) - abs(mean(signal))`.
- $\implies$ the smaller the *scaling factor*.
- the less sensitive the scaling factor is to the mean of the raw signal.

The smaller the sd,

- the the smaller `mean(abs(signal)) - abs(mean(signal))`.
- $\implies$ the bigger the *scaling factor*.
- $\implies$ the bigger the scaled signal.
- the more sensitive the scaling factor is to the mean of the raw signal.

When the mean of the raw signal is *not* close to zero:

- `mean(abs(signal))` and `abs(mean(signal))` are effectively equal for *small* sd.

When the mean of the raw signal is 0,

- the *scaling factor* is inverse proportional to sd.

The bigger the absolute mean of the signal,

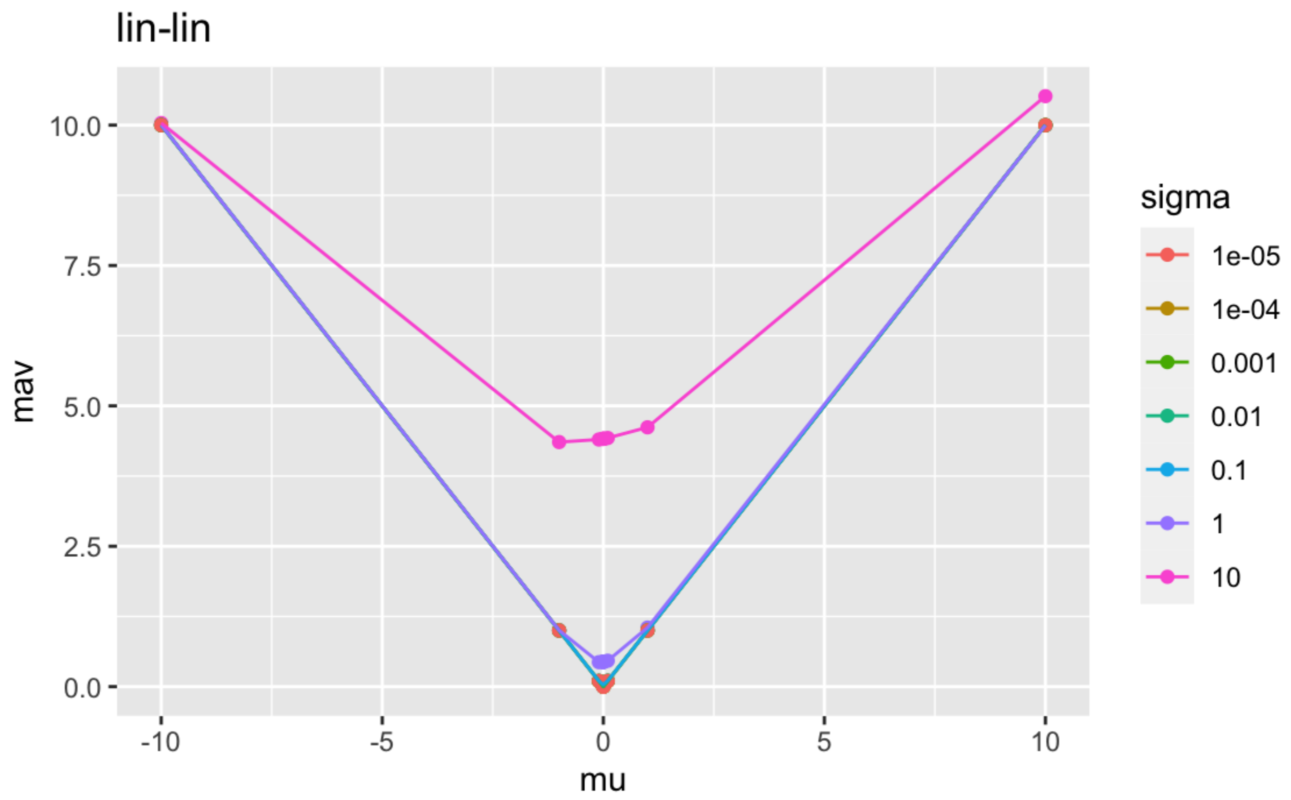
- the less sensitive the *scaling factor* is to sd.

Notice:

- Doubling the standard deviation of a mean zero Gaussian will also double it's average absolute value.

- The *expected absolute value* of a Gaussian is equal to `sd * sqrt(2/pi)`.
- If the mean isn't zero (which for slow trend following and carry is very likely) then the standard deviation will be biased downwards compared to abs() although that would be easy to fix by using MAD rather than STDEV.
- These points are from this discussion:
  - https://qoppac.blogspot.com/2016/01/pysystemtrader-estimated-forecast.html
  - (CarverRobert_SystematicTrading#F6: Price Volatility as Exponentially Weighted Moving Average (EWMA))

Signal follows a skewed t-distribution



Conclusion

- It looks like the values of μ and σ are much bigger concerns than skewness.

References

1. RMarkdown notebook:
   - git/rsystrade/misc/dev/Volatility targeting.Rmd
2. Questions and ToDo

---

1. See RMarkdown notebook in References↩