Turnover

****** TL;DR - see sections vi) and ix) ******

Contents:

i) Definitions (pp.1)

ii) SRFA vs Enterprise calculations (pp.1-2)

iii) Why SRFA turnover values are inaccurate (pp.2)

iv) Calculating SRFA's (transactional) turnover from scratch (pp.2)

v) Calculating Enterprise's (synthetic) turnover from scratch (pp.2-3)

vi) Monthly Audit (pp.3-4)

vii) Moving Average: calculation (pp.5-6)

viii) Moving Average: SQL (pp.6-8)

ix) Moving Average: interpretation (pp.8-9)

i) Definitions

Turnover represents the change of a portfolio's holdings between two given periods. Typically this is calculated with a monthly interval and for our purposes, this measure is almost exclusively used.

It is calculated by summing the absolute value changes in holdings from one month to another and then dividing the sum by two. The total is divided by two so as to not double count transactions.

For example, if I held only two stocks in my portfolio, Apple and Microsoft, then if Period 1 is 40% Apple and 60% Microsoft and Period 2 is 50% Apple and 50% Microsoft, then turnover would be 10%, because I've effectively just taken 10% from Microsoft and transferred it into Apple's portion of the holdings. The simplest intuition behind this is that rises and falls in holding positions do not occur independently of each other, but for our purposes are opposite sides of the same transaction.

Having some turnover month-to-month is expected, but if it diverges excessively from previous values, it could signal poor data quality. In other words, the underlying holdings have been loaded incorrectly.

Generally, whilst higher levels of turnover are more likely to signal poor data quality, they are not a guarantee of it. What the turnover is now relative to what it has been in the past is a more reliable indicator of data quality, which is why we use a Moving Average as part of our auditing (more on this below).

ii) SRFA vs Enterprise calculations

Turnover observed in SRFA is rarely, but mostly not, different from corresponding calculations made in client environments. In short, this is because Enterprise takes stock movements into account, whereas SRFA simply compares the raw holdings from one month to the next. In addition Enterprise's automated stock matching may differ from SRFA's.

In full, if we have two periods of stock positions, T_0 and T_1, then Enterprise creates a hypothetical position, T^H_0, that represents T_0's unchanged positions, but at T_1's prices. The turnover we view in client environments refers to the summed absolute differences between T^H_0 and T_1 divided by two, meaning if there is significant price volatility (whose data can be provided by the Market Data team in terms of individual securities) or stock mismatching (that isn’t reflected in the raw holdings files and therefore the FTP export files), there’ll be a discrepancy between the 'transactional' turnover we calculate internally (summed absolute differences between T_0 and T_1 divided by two) and the turnover calculated in Enterprise. Stock mismatching refers to when Enterprise matches stocks to positions that were not reflected in the underlying holdings files.

For example, Apple may be incorrectly matched to Apple's ADR position and since these are treated as separate positions, turnover will be influenced by the sum of these two positions divided by two, which is more likely to be significantly higher than the absolute difference between the two Apple positions divided by two, thereby inflating reported turnover in the client's environment.

iii) Why SRFA turnover values are inaccurate

This is because it orders snapshots according to when they were loaded, not according to their actual chronology. For example, for a given fund, if you loaded January, then March, then February, then March's turnover will signify the change from January to March instead of February to March. This is a global problem because the order of loading isn't typically chronological: even if it was ordered in the input folder, the order would break down once funds start invalidating.

iv) Calculating SRFA's (transactional) turnover from scratch

Keep the first 7 columns of your sheet blank. Columns A and B should contain Period 1's stock positions and weights, respectively. Columns E and F should contain Period 2's stock positions and weights, respectively.

In Column C, lookup Period 1's stock positions on Period 2's weights. In Column G, lookup Period 2's stock positions on Period 1's weights. Then, copy the sheet and paste values.

Filter Column G on #N/A and paste these values directly underneath Period 1's values. Then replace all #N/As with 0s.

In Column D, input the following: =ABS(B2-C2), then drag this down and find the total and divide by two; this is the turnover value. Then, find the total from Column B's weights and divide the turnover value by this; this is the turnover percentage.

v) Calculating Enterprise's (synthetic) turnover from scratch

Performing this calculation assumes that there hasn't been mismatching from SRFA's reported positions to the Enterprise database.

Since we're considering price movements, we'd need to find the first period's hypothetical set of weights that represents the first period's positions, but in the second period's prices. To do this, we need the prices and volumes of each of the first and second period's positions. From this, we can calculate the NAVs (net asset values: summed products of volume and price for each position) for each of the periods, in addition to the hypothetical NAVs, which will lead directly to the weights for the first, second and hypothetical periods (W1, W2, HypW1).

'Transactional' turnover refers to the change from W1 to W2. 'Synthetic' turnover refers to the change from HypW1 to W2. The difference between the two values is explained below:

Transactional Turnover = Synthetic Turnover + (Intrasynthetic Turnover - Overlap)

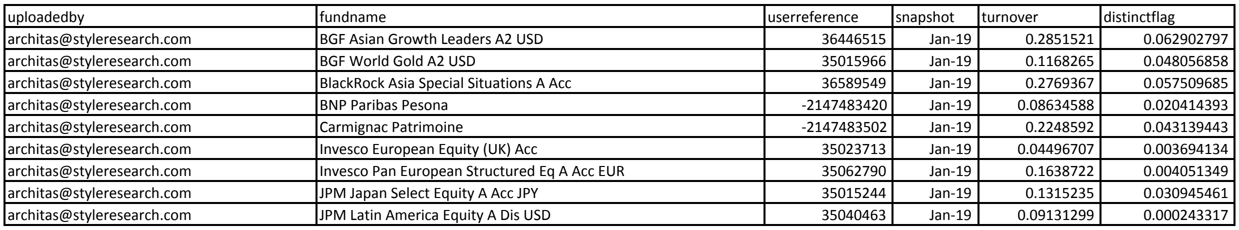
'Intrasynthetic Turnover' refers to the change from W1 to HypW1 and 'Overlap' refers to the change accounted for by both Synthetic and Intrasynthetic Turnover, meaning the term above in parentheses represents the change exclusively accounted for by comparing W1 to HypW1. If you wish to isolate either Synthetic or Intrasynthetic Turnover, the parenthesised term can be switched accordingly.

vi) Monthly Audit

Each month, typically within the first week, a full turnover audit of all active client environments is performed, using the Moving Average test (see section vii)).

Results are generated from SQL via the Enterprise database, using the script *Private Lane - Moving Average Turnover Audit.sql*.

For example, below is the set of flags for January 2019. Results are separated according to client and may duplicate across them.



uploadedby	fundname	userreference	snapshot	turnover	distinctflag
architas@styleresearch.com	BGF Asian Growth Leaders A2 USD	36446515	Jan-19	0.2851521	0.062902797
architas@styleresearch.com	BGF World Gold A2 USD	35015966	Jan-19	0.1168265	0.048056858
architas@styleresearch.com	BlackRock Asia Special Situations A Acc	36589549	Jan-19	0.2769367	0.057509685
architas@styleresearch.com	BNP Paribas Pesona	-2147483420	Jan-19	0.08634588	0.020414393
architas@styleresearch.com	Carmignac Patrimoine	-2147483502	Jan-19	0.2248592	0.043139443
architas@styleresearch.com	Invesco European Equity (UK) Acc	35023713	Jan-19	0.04496707	0.003694134
architas@styleresearch.com	Invesco Pan European Structured Eq A Acc EUR	35062790	Jan-19	0.1638722	0.004051349
architas@styleresearch.com	JPM Japan Select Equity A Acc JPY	35015244	Jan-19	0.1315235	0.030945461
architas@styleresearch.com	JPM Latin America Equity A Dis USD	35040463	Jan-19	0.09131299	0.000243317

architas@styleresearch.com	Kames Property Income Fund	-2147483441	Jan-19	0.1899512	0.102512305
architas@styleresearch.com	Ossiam Shiller Barclays CAPE US Sector Val TR 1CU	36680523	Jan-19	0.5471118	0.056376534
architas@styleresearch.com	Pictet-US Equity Selection-P USD	35195179	Jan-19	0.1429185	0.00957217
architas@styleresearch.com	Schroder ISF Greater China A Acc	35081916	Jan-19	0.1126242	0.020203257
datamanager@cazenovecapital.com	BBH Luxembourg Funds - Core Select X	36399462	Jan-19	0.1921831	0.038329993
datamanager@cazenovecapital.com	BGF Asian Dragon A2 USD	35016014	Jan-19	0.1024223	0.010964759
datamanager@cazenovecapital.com	BGF Asian Growth Leaders A2 USD	36446515	Jan-19	0.311257	0.089007753
datamanager@cazenovecapital.com	Equitile RESILIENCE	-2147483396	Jan-19	0.3202727	0.078954958
datamanager@cazenovecapital.com	FP Argonaut European Alpha GBP A Acc	36415956	Jan-19	0.954475	0.378000968
datamanager@cazenovecapital.com	Invesco Asian Equity A Annual Dist USD	35017196	Jan-19	0.3069001	0.169070219
datamanager@cazenovecapital.com	M&G Global Emerging Markets GBP A Inc	35996098	Jan-19	0.1250881	0.049116602
datamanager@cazenovecapital.com	UBS ETF MSCI World Socially Resp UCITS (USD) Ad	36362211	Jan-19	0.3287011	0.041908691
datamanager@coutts.com	BGF Asian Dragon A2 USD	35016014	Jan-19	0.09931651	0.007858985
datamanager@coutts.com	Invesco European Equity (UK) Acc	35023713	Jan-19	0.04496707	0.003694134
datamanager@coutts.com	iShares MSCI Japan GBP Hedged UCITS ETF (Acc)	36468439	Jan-19	0.06229435	0.001047035
datamanager@coutts.com	Schroder ISF China Opportunities A Acc	35182777	Jan-19	0.1016851	0.010985272
datamanager@coutts.com	Schroder ISF Greater China A Acc	35081916	Jan-19	0.1126055	0.020184497
datamanager@coutts.com	UBS ETF MSCI World Socially Resp UCITS (USD) Ad	36362211	Jan-19	0.3287011	0.041908691
datamanager@coutts.com	Wellington US Research Equity S	35062584	Jan-19	0.1562401	0.048424945
datamanager@standardlife.com	Aberdeen Asia Pacific Equity Enhanced Index Fund	-2147483413	Jan-19	0.08567737	0.020594894
datamanager@standardlife.com	Aberdeen North American Equity EQAME	30006496	Jan-19	0.1570948	0.039116497
datamanager@standardlife.com	BGF Asian Dragon A2 USD	35016014	Jan-19	0.1011296	0.009672121
datamanager@standardlife.com	BGF Asian Growth Leaders A2 USD	36446515	Jan-19	0.2942091	0.071959842
datamanager@standardlife.com	BGF Latin American A2 USD	35016056	Jan-19	0.1073099	0.018644219
datamanager@standardlife.com	Invesco European Equity (UK) Acc	35023713	Jan-19	0.04496707	0.003694134
datamanager@standardlife.com	Invesco Global Emerging Markets (UK) Acc	35022879	Jan-19	0.2035259	0.072512122
datamanager@standardlife.com	iShares UK Equity Index (UK) L Acc	35169938	Jan-19	0.03224604	0.010807198
datamanager@standardlife.com	JPM Emerging Markets Dividend A Acc USD	36474061	Jan-19	0.1136866	0.022729913
datamanager@standardlife.com	Royal London UK Smaller Companies Fund	35731288	Jan-19	0.05673529	0.00560125
datamanager@standardlife.com	T Rowe European Equity A EUR	35805576	Jan-19	0.07942887	0.00335332
datamanager@unilever.com	*UNIVEST FCP - GA - Fidelity	-2147483591	Jan-19	0.08270689	0.000486202
datamanager@unilever.com	*UNIVEST FCP - GA - Robeco AP	-2147483592	Jan-19	0.0570603	0.026684364
jpmorganpb@styleresearch.com	BBH Luxembourg Funds - Core Select X	36399462	Jan-19	0.1921831	0.038329993
jpmorganpb@styleresearch.com	BGF Asian Dragon A2 USD	35016014	Jan-19	0.1011247	0.009667188
jpmorganpb@styleresearch.com	JPM India A Dis USD	35019824	Jan-19	0.04504158	0.005240828
jpmorganpb@styleresearch.com	Wellington US Research Equity S	35062584	Jan-19	0.1562401	0.048424945
quilter@styleresearch.com	BGF Asian Growth Leaders A2 USD	36446515	Jan-19	0.3171857	0.094936449
quilter@styleresearch.com	Invesco European Equity (UK) Acc	35023713	Jan-19	0.04496707	0.003694134
quilter@styleresearch.com	Invesco Income (UK) Inc	35023721	Jan-19	0.06020721	0.010798018
quilter@styleresearch.com	iShares MSCI EM Latin America UCITS ETF (Acc)	36259536	Jan-19	0.04052235	0.002608969
quilter@styleresearch.com	JOHCM Japan B GBP	35098964	Jan-19	0.277593	0.1237096
srpa_ent@schrodermultimanager.com	BGF Asian Dragon A2 USD	35016014	Jan-19	0.1011247	0.009667188
srpa_ent@schrodermultimanager.com	Invesco European Equity (UK) Acc	35023713	Jan-19	0.04496707	0.003694134
srpa_ent@schrodermultimanager.com	Invesco Income (UK) Inc	35023721	Jan-19	0.06020721	0.010798018

The above results would be investigated and classified as a genuine data error, a false signal or with a currently unknown status (see section ix)).

In addition, a KPI (key performance indicator) is published. $\text{KPI}_{\text{turnover}}$ measures the percentage of snapshots for which there is potentially a data error. As such, the numerator would be the total amount of flags identified as genuine data errors or with currently unknown statuses. The denominator would be the total amount of funds for all clients; this would include duplicates across clients since the audit distinguishes between client-specific snapshots relating to the same funds.

vii) Moving Average: calculation

This is a particular test used by the Fund Data Team to indicate whether or not a snapshot has poor data quality. Enterprise turnover values are considered for this test, since they are what clients observe and so are assumed to be the most direct indication of data quality. For a given set of snapshots' turnovers, the Moving Average is defined as the sum of the mean and the product of the weighted standard deviation and a certain parameter. In full:

$\mathbf{MA = \mu + A_{SD}\sigma_{wtd}}$, where $1.75<A_{SD}<2.25$ and

$\mathbf{\sigma_{wtd} = A_S\sigma + D_{t-1}((1-A_S)/2) + D_{t-2}((1-A_S)/2)}$, where $0.75<A_S<0.85$ and where

$\mathbf{A_{SD} = 1.75 + 0.5((rank-1)/(count-1))}$ and $\mathbf{A_S = 0.75 + 0.1((rank-1)/(count-1))}$

MA = Moving Average

μ = mean

A_{SD} = adaptive standard deviation parameter

σ_{wtd} = weighted standard deviation

A_S = adaptive smoothing parameter

D_{t-1} = turnover preceding the value under consideration

D_{t-2} = turnover preceding D_{t-1}

If any given turnover value is greater than its corresponding moving average (calculated for each fund), then it is flagged. A_{SD} and A_S vary for each fund, whereas D_{t-1} and D_{t-2} vary for each snapshot's turnover. We exclude values below 5% because across repeated observations, snapshots with these values are highly unlikely to be incorrectly loaded and are very high in quantity.

The standard deviation is weighted by including preceding values (D_{t-1} and D_{t-2}). It is adaptive (varies for each fund) in two ways: i) the amount of standard deviations by which the Moving Average differs from the mean varies for each fund and ii) the amount of weight placed on the standard deviation compared to the preceding turnover values varies for each fund.

Funds are ranked according to their standard deviation - as per the above definition, the lower the rank, the higher the standard deviation.

A_{SD} measures the strictness of the Moving Average test for a given snapshot's turnover. It represents how many standard deviations away the Moving Average is from the mean. The 'how many' aspect depends on the volatility of the fund itself with regards to turnover. If it is particularly volatile, the Moving Average is more sensitive to flagging and so A_{SD} is lower and vice versa.

A_S refers to the extent to which the Moving Average relies on the overall data-set rather than more recent snapshots. A perfectly smooth parameter would have a value of 1. Having the parameter be less than 1 is beneficial due to greater versatility; more reliance on recent snapshots means less feeder data is required to produce results. However, if the parameter is too low, results may become unreliable. The parameter 'adapts' within the range specified above since it was observed that that produced the most flagged results without uniformly compromising on their reliability; in this way it adds value through improving the predictive power of the overall test (across all funds) as opposed to individual tests. On a marginal level, there is no reason to vary the parameter (instead it can be fixed at 0.8, or 1 if there are no issues regarding data access).

viii) Moving Average: SQL

Below is the following SQL script used to generate flagged turnover results (the table names will vary for each test) - comments are provided:

```
create table #calc (fundname varchar(100), userreference varchar(100), avgturnover
float, stdevturnover float, ASDP float, ASP float)

insert into #calc (fundname, userreference, avgturnover, stdevturnover, ASDP, ASP)

select fundname, userreference, avg(turnover) as avgturnover, stdevp(turnover) as
stdevturnover,

(1.75+0.5*(cast((row_number() over (order by stdevp(turnover))) as float)/(select
count(distinct fundname) from dbo.FinalResults_ToDelete_2017Jan01_2019Mar04))) ASDP,
--ASDP = adaptive standard deviation parameter

(0.75+0.1*(cast((row_number() over (order by stdevp(turnover))) as float)/(select
count(distinct fundname) from dbo.FinalResults_ToDelete_2017Jan01_2019Mar04))) ASP
--ASP = adaptive smoothing parameter

from dbo.FinalResults_ToDelete_2017Jan01_2019Mar04 --changing variable (found elsewhere
too)

group by fundname, userreference


create table #past (fundname varchar(100), userreference varchar(100), uploadedby
varchar(100), snapshotdate varchar(100), prev float, prevprev float)

insert into #past (fundname, userreference, uploadedby, snapshotdate, prev, prevprev)

select fundname, userreference, uploadedby, snapshotdate,

isnull(LAG(turnover) over(partition by
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.uploadedby,
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.fundname

order by
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.SnapshotDate),(isnull(LAG(turnover,2)
over(partition by dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.uploadedby,
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.fundname

order by
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.SnapshotDate),(isnull(LAG(turnover,3)
over(partition by dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.uploadedby,
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.fundname
order by dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.SnapshotDate),0))))) as prev,
--recursive isnull function gives lag2 if lag1 is 0, given a range of 3 snapshotdates
(lag1 to lag3)

isnull(LAG(turnover,2) over(partition by
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.uploadedby,
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.fundname

order by
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.SnapshotDate),(isnull(LAG(turnover,3)
over(partition by dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.uploadedby,
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.fundname

order by
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.SnapshotDate),(isnull(LAG(turnover,4)
over(partition by dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.uploadedby,
dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.fundname

order by dbo.FinalResults_ToDelete_2017Jan01_2019Mar04.SnapshotDate),0))))) as prevprev
--recursive isnull function gives lag3 if lag2 is 0, given a range of 3 snapshotdates
(lag2 to lag4)

from dbo.FinalResults_ToDelete_2017Jan01_2019Mar04

group by fundname, userreference, uploadedby, snapshotdate, turnover



select uploadedby, fundname, userreference, snapshot, turnover, max(madiff) as
distinctflag --distinctflag filters for results where turnovers exceed moving averages:

from(select db.uploadedby, db.fundname, db.userreference, db.snapshot, db.turnover,
              --their only significance is in being >0

db.turnover-(avgturnover+ASDP*((ASP*stdevturnover)+prev*((1-ASP)/2)+prevprev*((1-ASP)/2
))) as madiff

from dbo.FinalResults_ToDelete_2017Jan01_2019Mar04 db

inner join #calc on #calc.fundname=db.fundname

inner join #past on #past.fundname=db.fundname

where snapshot = 'Jan-2019' --specify snapshot

and
db.turnover-(avgturnover+ASDP*((ASP*stdevturnover)+prev*((1-ASP)/2)+prevprev*((1-ASP)/2
)))>0) a

group by uploadedby, fundname, userreference, snapshot, turnover



drop table #calc
drop table #past
```

ix) Moving Average: interpretation

The below set of suggestions is by no means exhaustive, but is a reliable starting point.

If you can verify that i) the right fund is being loaded and ii) the holdings are being loaded properly, then there is nothing problematic about a snapshot's data quality, even if it is flagged. Verifying condition i) requires investigating the underlying file behind the holdings and the Lipper mapping on SRFA via the supplied fund name found in the file. An occurent issue related to this condition is the incorrect mapping of similarly named funds; this should be at the forefront of investigations relating to i).

If we assume that there is approximate parity between Enterprise's and SRFA's turnover values, then calculating SRFA's turnover from scratch (and observing the difference between the two - note that based on the above assumption, a few percentage points' difference is negligible) will be a reliable indication of whether or not the snapshot's holdings are being loaded properly. A shorcut in this regard is to go to *Funds Overview* on SRFA and observe the turnovers directly. If the flagged snapshot and the one preceding it were loaded consecutively, then that value need not be calculated from scratch. Practices from regular turnover audits will apply when verifying ii), for example, assessing load specifications.

Viewing a fund's *Analyses* under *Explore* in the client's environment will provide a useful audit trail; both in terms of snapshots and when respective analyses were run, in addition to whether or not analyses overwrote preceding ones. It may well be that one flag uncovers a far more prevalent issue! Moreover, the Maths Development and/or Market Data Teams should be able to assist if you need to see exactly how Enterprise calculated a snapshot's turnover.

Turnover cannot be over 100% by definition, since a portfolio only 'occupies' that 100%. If a turnover value 'spikes' to an abnormally high value, with subsequent values 'appearing' normal, this would indicate that there was a data error in the past that is persisting into the present. If, however, a turnover value 'spikes' to an abnormally high value, with a subsequent 'correction' of an abnormally high value, this indicates that there was a data error in the past that has 'corrected itself' and is no longer persistent into the present.

It is crucial that the quality of the feeder data allows for the test to generate appropriate flags. If past and present data errors have not been rectified and are included in future tests, values that typically would be considered as abnormal will be 'normalised'. Correcting for this may seem like an impossible task, however, if past values are independently tested as final values incrementally, any dormant data errors may be discovered and corrected. For example, if March 2019 is the latest snapshot, then if you run a Moving Average test from November 2017 to October 2018, October 2018 would be considered in reference to the past 11 snapshots from that point and not in reference to the past 6 snapshots if the test from April 2018 to March 2019 was considered.

This approach only works if past data sets are a reliable frame of reference. The more likely eventuality is that the entirety of data errors will not all be uncovered at once, but will be rectified on an ongoing basis. Assuming genuine data errors are rectified as and when they occur (each month), overall data quality will improve over time without having to pre-emptively correct the past.

Flags are interpreted as being caused by a data error, being a false signal, or having a currently unknown status. The most common reason for a false signal is because a fund is active and so has consistently high turnover values, however this should largely be controlled for by the Moving Average. If the flag's status is currently unknown, referral to Maths Development, Market Data or directly to the Fund Manager (this is not a recommended first option) is the next step.