

Finding Structure in Brownian Motion through Correspondence Analysis

Fionn Murtagh

School of Computer Science, Queen's University Belfast
Belfast BT7 1NN, Northern Ireland, UK

Email f.murtagh@qub.ac.uk

August 5, 2003

Abstract

We study the use of categorical or qualitative data coding, as used commonly in correspondence analysis, for finding faint structure in financial time series. The overall objective is to use faint patterns in such data streams for prediction. We recall relevant definitions from correspondence analysis, in particular the simultaneous spatial and clustering analysis which is facilitated by it. We study in some depth a data set of financial futures (daily highs) in order to show that this approach to faint pattern finding, at an appropriate resolution level, works very well in practice.

Keywords: data coding, correspondence analysis, minimum variance hierarchical clustering, efficient market hypothesis, geometric Brownian motion, financial modeling, time series prediction, data aggregation, data resolution.

1 Introduction

Correspondence analysis is a data analysis approach based on low-dimensional spatial projection. Unlike other such approaches, it particularly well caters for qualitative or categorical input data. Often, cluster analysis is also carried out, and the low-dimensional output representation contrasted with sets of clusters found. Input data coding impacts directly on the analysis carried out. Therefore input data coding becomes quite important when this data analysis approach is used. Examples of input data coding, which we will not discuss further here, include: doubling, complete disjunctive form, fuzzy coding, personal equation, double rescaling.

Our objectives in this analysis are to take data recoding as proposed in Ross (2003) and study it as a type of coding commonly used in correspondence analysis. Ross (2003) uses input data recoding to find faint patterns in otherwise apparently structureless data. The implications of doing this are important: we

wish to know if such data recoding can be applied in general to apparently structureless financial or other data streams.

In this article, our objectives are as follows.

1. Using categorical or qualitative coding may allow structure, imperceptible with quantitative data, to be discovered.
2. Quantile-based categorical coding (i.e., the uniform prior case) has beneficial properties.
3. An appropriate coding granularity, or scale of problem representation, should be sought.
4. In the case of a time-varying data signal (which also holds for spatial data, *mutatis mutandis*) non-respect of stationarity should be checked for: the consistency of our results will inform us about stationarity present in our data.
5. Structures (or models or associations or relationships) found in training data are validated on unseen test data. But if a data set consistently supports or respects these structures then *a fortiori* leaving- k -out cross-validation is achieved.
6. Departure from average behavior is made easy in the analysis framework adopted. This amounts to fingerprinting the data, i.e. determining patterns in the data that are characteristic of it.

2 Brownian Motion

2.1 Efficient Market Hypothesis and Geometric Brownian Motion

The efficient market hypothesis was formulated initially by Samuelson (1965): if y_t is the value of a financial asset, then the expected value at time $t + 1$ is related to previous values as follows.

$$E\{y_{t+1} \mid y_0, y_1, \dots, y_t\} = y_t$$

When stochastic processes satisfy this conditional probability, they are termed martingales (Doob, 1953). The efficient market hypothesis is taken as due to rational behavior and market efficiency. A martingale is informally a model of a fair game in that wins and losses become equal over time. An implication of the efficient market hypothesis is that price changes are not predictable from a historical time series of these prices. Empirical evidence supports the efficient market hypothesis, although Mantegna and Stanley (2000) report that the additional use of fundamentals such as earnings/price ratios, dividend yields, and term-structure variables allow for predictions on a longer time horizon.

Differenced values of the time series with constant time steps are studied through Brownian motion: for $0 \leq i < \infty$, the variable $y_{t+1} - y_t$ is independent of all $y_i, i < t$, and follows a Gaussian distribution. As in the efficient market hypothesis, in Brownian motion a future price depends only on the present price, and not at all on the past prices. Furthermore in Brownian motion, price difference is Gaussian. Ross (2003) points to two problems with the use of Brownian motion to analyze financial data streams: firstly, use of a Gaussian implies the need for negative prices; and, secondly, it seems unrealistic to expect that a given gain or loss $y_{t+1} - y_t$ occurs with the same probability irrespective of whether y_t is large or small.

These difficulties with Brownian motion in financial time series are avoided with geometric Brownian motion. In geometric Brownian motion, the variable y_{t+1}/y_t is not dependent on any $y_i, i < t$, and $\log(y_{t+1}/y_t)$ is Gaussian. Therefore the ratio of price y_{t+1} to present price y_t follows a lognormal distribution, and is independent of all past prices. With drift μ and volatility σ , geometric Brownian motion satisfies $E\{y_t\} = y_0 \exp t(\mu + \sigma^2/2)$.

2.2 Data Transformation and Coding

Using crude oil data, Ross (2003) shows how structure can be found in apparently geometric Brownian motion, through data recoding. Considering monthly oil price values, $P(i)$, and then $L(i) = \log(P(i))$, and finally $D(i) = L(i) - L(i-1)$, a histogram of $D(i)$ for all i should approximate a Gaussian. The following recoding, though, gives rise to a somewhat different picture: response categories or states 1, 2, 3, 4 are used for values of $D(i)$ less than or equal to -0.01 , between the latter and 0, from 0 to 0.01, and greater than the latter. Then a cross-tabulation of states 1 through 4 for y_{t+1} , against states 1 through 4 for y_t , is determined. The cross-tabulation can be expressed as a percentage. Under geometric Brownian motion, one would expect constant percentages. This is not what is found. Instead there is appreciable structure in the contingency table.

Ross (2003) pursues exploration of geometric Brownian motion basis of Black-Scholes option cost. States-based pricing leads to greater precision compared to a one-state alternative. The number of states is left open with both a 4-state and a 6-state analysis discussed (Ross, 2003, chap. 12). A χ^2 test of independence of the contingency table from a product of marginals (cf. discussion in regard to correspondence analysis, to follow) is used with degrees of freedom associated with contingency table row and column dimensions: this provides a measure of how much structure we have, but not between alternative contingency tables. The total inertia or trace of the data table grows with contingency table dimensionality, so that is of no help to us either. For the futures data used below (see Figure 1), and contingency tables of size 3×3 , 4×4 , 5×5 , 6×6 , and 10×10 , we find traces of value: 0.0118, 0.0268, 0.0275, 0.0493, and 0.0681, respectively. Barring the presence of low-dimensional patterns arising in such a sequence of contingency tables, we will *always* find that greater dimensionality implies greater complexity (quantified, e.g., by trace) and therefore structure.

To address the issue of number of coding states to use, in order to search for latent structure in such data, one approach that seems very reasonable is to explore the dependencies and associations based on fine-grained structure; and include in this exploration the possible aggregation of the fine-grained states.

3 Basics of Correspondence Analysis

Correspondence analysis (Benzécri, 1976; 1992) allows us to analyze contingency table structure. Part and parcel of this analysis is a simultaneous clustering of rows and columns of the contingency table and, with user interaction, determining appropriate resolution level or granularity of input data coding.

The moment of inertia of a cloud of points in a Euclidean space, with both distances and masses defined, is the sum for all elements of I of the products of mass by distance squared from the center of the cloud:

$$M^2(N_J(I)) = \sum_{i \in I} f_i \|f_J^i - f_J\|_{f_J}^2 = \sum_{i \in I} f_i \rho^2(i) \quad (1)$$

We will explain these terms in turn. The term, ρ , is the Euclidean distance from the cloud center, and f_i is the mass of element i . Let us take a step back: the given contingency table data are denoted $k_{IJ} = \{k_{IJ}(i, j) = k(i, j); i \in I, j \in J\}$. We have $k(i) = \sum_{j \in J} k(i, j)$. Analogously $k(j)$ is defined, and $k = \sum_{i \in I, j \in J} k(i, j)$. Next, $f_{IJ} = \{f_{ij} = k(i, j)/k; i \in I, j \in J\} \subset \mathbb{R}_{I \times J}$. Similarly f_I is defined as $\{f_i = k(i)/k; i \in I, j \in J\} \subset \mathbb{R}_I$, and f_J analogously.

Next back to the first right hand side term in eqn. 1. The conditional distribution of f_J knowing $i \in I$, also termed the j th profile with coordinates indexed by the elements of I , is

$$f_J^i = \{f_j^i = f_{ij}/f_i = (k_{ij}/k)/(k_i/k); f_i \neq 0; j \in J\}$$

and likewise for f_I^j .

We can further rationalize our notation by considering a tensor calculus of transitions between probability spaces. A transition from I to J is an element of the tensor product $\mathbb{R}_J \otimes \mathbb{R}^I$. It is a function on I , but with values in the J measures; or the conditional probability of j given i . Such a transition takes masses (or probability measures or densities) from I to J ; and associates every function on J with a function on I .

The cloud of points consists of the couple: profile coordinate and mass. We have $N_J(I) = \{(f_J^i, f_i); i \in I\} \subset \mathbb{R}_J$, and again similarly for $N_I(J)$.

From eqn. 1, it can be shown that

$$M^2(N_J(I)) = M^2(N_I(J)) = \|f_{IJ} - f_I f_J\|_{f_I f_J}^2 = \sum_{i \in I, j \in J} (f_{ij} - f_i f_j)^2 / f_i f_j \quad (2)$$

The term $\|f_{IJ} - f_I f_J\|_{f_I f_J}^2$ is the χ^2 metric between the probability distribution f_{IJ} and the product of marginal distributions $f_I f_J$, with as center of the metric the product $f_I f_J$.

In correspondence analysis, the choice of χ^2 metric of center f_J is linked to the *principle of distributional equivalence*, explained as follows. Consider two elements j_1 and j_2 of J with identical profiles: i.e. $f_I^{j_1} = f_I^{j_2}$. Consider now that elements (or columns) j_1 and j_2 are replaced with a new element j_s such that the new coordinates are aggregated profiles, $f_{ij_s} = f_{ij_1} + f_{ij_2}$, and the new masses are similarly aggregated: $f_{j_s} = f_{j_1} + f_{j_2}$. Then there is *no effect* on the distribution of distances between elements of I . The distance between elements of J , other than j_1 and j_2 , is naturally not modified. This description has followed closely Jambu (1978, chap. 2).

The principle of distributional equivalence leads to representational self-similarity: aggregation of rows or columns, as defined above, leads to the same analysis. Therefore it is very appropriate to analyze a contingency table with fine granularity, and seek in the analysis to merge rows or columns, through aggregation.

A further implication of the principle of distribution equivalence is related to the weight distribution. If, in our analysis of profiles j_1 and j_2 above, the masses f_{j_1} and f_{j_2} are equal, then equal profiles implies $f_{ij_1} = f_{ij_2}$ rather than $f_{ij_1}/f_{j_1} = f_{ij_2}/f_{j_2}$. Our profiles are then more immediately and directly furnished by the original data. In addition we are less likely to have disproportionate influence by one element vis-à-vis another, due to imbalance of masses.

One way to arrange for roughly equal masses is to define our coding by means of frequency quantiles. For example, by choosing the median variable (coordinate) value and assigning all values greater than or equal to the median to 1, and all values less than the median (50th quantile) to zero, then axiomatically each column of the coded values will sum to $n/2$, where $n = |I|$, the set cardinality of I . Without loss of generality, we will ignore discretization effects for small n . Proceeding further along these lines, we can obtain a 10-valued binary coding by using 10th, 20th, ... quantiles. The principle of data coding using quantiles was used effectively in Murtagh and Sarazin (1993) and Murtagh, Aussem & Sarazin (1995).

In terms of Bayes rule, $P(j|i) = P(i|j)P(j)/P(i)$. From the quantile-based coding we have uniform priors: $p(j) = \text{constant}$. Therefore, with the notation used above and with constant k , $f_j^i = k f_i^j / f_i$. This is seen to be verified if we write $f_j^i = f_{ij} / f_i$, $f_i^j = f_{ij} / f_j$, and (as always) require that $f_i, f_j \neq 0$.

4 Hierarchical Classification

Hierarchical agglomeration on n observation vectors, $i \in I$, involves a series of $1, 2, \dots, n - 1$ pairwise agglomerations of observations or clusters, with the following properties. A hierarchy $H = \{q | q \in 2^I\}$ such that (i) $I \in H$, (ii) $i \in H \forall i$, and (iii) for each $q \in H, q' \in H : q \cap q' \neq \emptyset \implies q \subset q'$ or $q' \subset q$. An indexed hierarchy is the pair (H, ν) where the positive function defined on H , i.e., $\nu : H \rightarrow \mathbb{R}^+$, satisfies: $\nu(i) = 0$ if $i \in H$ is a singleton; and (ii) $q \subset q' \implies \nu(q) < \nu(q')$. Function ν is the agglomeration level. Take $q \subset q'$, let $q \subset q''$ and $q' \subset q''$, and let q'' be the lowest level cluster for which this is true.

Then if we define $D(q, q') = \nu(q'')$, D is an ultrametric. In practice, we start with a Euclidean or other dissimilarity, use some criterion such as minimizing the change in variance resulting from the agglomerations, and then define $\nu(q)$ as the dissimilarity associated with the agglomeration carried out.

For Euclidean distance inputs, the following definitions hold for the minimum variance or Ward error sum of squares agglomerative criterion (Murtagh, 1985):

- Coordinates of the new cluster center, following agglomeration of q and q' , where m_q is the mass of cluster q defined as cluster cardinality, and (vector) q denotes using overloaded notation the center of (set) cluster q : $q'' = (m_q q + m_{q'} q') / (m_q + m_{q'})$.
- Following the agglomeration of q and q' , we define the following dissimilarity: $(m_q m_{q'}) / (m_q + m_{q'}) \|q - q'\|^2$.

These two definitions are all we need to specify the hierarchical clustering algorithm. When q and q' are both singletons, the latter rule implies that a weighting of 0.5 is applied to the Euclidean distance. Hierarchical clustering based on factor projections, if desired using a limited number of factors (e.g. 7) in order to filter out the most useful information in our data, provides for a consistent framework. In such a case, hierarchical clustering can be seen to be a mapping of Euclidean distances into ultrametric distances.

5 Granularity of Coding

As noted, we use

1. quantile coding motivated (i) by the desire on our part to find structure in Brownian motion signals, and (ii) by the fact that it lends itself well (in that it furnishes a uniform mass density) to the analysis and display properties of correspondence analysis; and
2. an overly fine-grained set of coding categories, so that a satisfactory outcome (a *satisficing* solution in scheduling terminology) is obtained by aggregating these categories.

The latter objective could be sought by many different clustering approaches. Such clustering approaches are often related though: for example Zha et al. (2001) show how data clustering through optimal graph decomposition is, in effect, one particular property of correspondence analysis. We will use correspondence analysis as an interactive analysis environment to address the following questions:

1. To aggregate the fine-resolution coding categories used, we need strongly associated coding categories.
2. Less influential coding categories are sought in order, possibly, to bypass them later in practical application.

3. In addition we will take into account possible non-stationarity over the time period of the data stream.
4. Generalizing the leaving- k -out approach to validation, we will seek consistency of results obtained for sub-intervals. If we can experimentally show that all possible sufficiently-sized sub-intervals of the time series manifest the same results, then *a fortiori* we are exemplifying how unseen data will behave.

To address point 3, and simultaneously point 4, we will take sets of 2500 values from the time series. Tables 1 through 4 show data to be analyzed, derived from time series values 1 to 2500 (identifier i), values 3001 to 5500 (identifier k), 2001 to 4500 (identifier m), and values 3600 to 6100 (identifier n).

Figure 2 shows the projections of the profiles in the plane of factors 1 and 2, using all four data tables shown in Tables 1–4. The result is very consistent: cf. how $\{i1, k1, m1, n1\}$ are tightly grouped, as are $\{i2, k2, m2, n2\}$, reasonably so $\{i10, k10, m10, n10\}$, and so on. The full space of all factors has to be used to verify the clustering seen in this planar (albeit least squares optimal) projection.

A clustering in a full coordinate space (7 factors used) allowed a 7-cluster solution to be obtained, – a solution that preceded a large increase in cluster agglomeration levels (indicated in Figure 3).

The clusters found are listed in Table 5. Contributions and correlations both measure relationship importance, and will be defined below in section 6.1. In cluster 65, coding category 9 is predominant. In cluster 68, coding categories 2 and 3 are predominant. Cluster 69 is mixed. Cluster 70 is dominated by coding category 10. In cluster 71, coding category 8 is predominant. Cluster 72 is defined by coding category 1. Finally, cluster 73 is dominated by coding category 5.

From the clustering, we provisionally retain coding categories 1; 2 and 3 together; 5; 8; 9; and 10. We flag response categories 4, 6, and 7 as being unclear and best avoided.

To check the coding relative to stationarity, Figure 4 shows that the global code boundaries are close to the time series sub-interval code boundaries.

6 Fingerprinting the Price Movements

6.1 Decomposition of Inertia: Contributions and Correlations

We have already looked at how correspondence analysis decomposes the inertia of clouds $N_J(I)$ or, analogously and closely related, $N_I(J)$. The moments of inertia, λ_α , $\alpha = 1, 2, \dots, \min\{n-1, m-1\}$ ($n = |I|$, $m = |J|$), are associated with factors. We could say that correspondence analysis maps a cloud (of profiles), in a space endowed with the χ^2 distance, into a cloud of points in a space endowed with the Euclidean distance. The projection of a point on a factor is denoted $F_\alpha(i)$.

$f_i F_\alpha^2(i)/\lambda_\alpha$ is the relative contribution of point i to the moment of inertia λ_α . (Often denoted CTR.)

Factors are determined by how much the elements contribute to their dispersion. Therefore the values of CTR are examined in order to identify or to name the factors (for example, with higher order concepts). (Informally, CTR allows us to work from the elements towards the factors.)

$\cos^2 a = F_\alpha^2(i)/\rho^2(i)$ is the relative contribution of the factor α to point i . (Often denoted COR.)

The values of COR are squared cosines, which can be considered as being like correlation coefficients. If $\text{COR}(i, \alpha)$ is large (say, around 0.8) then we can say that that element is well explained by the axis of rank α . (Informally, COR allows us to work from the factors towards the elements.)

The analysis of clusters in terms of factors and vice versa is carried out by programme VACOR (see Benzécri, 1992). We used our implementation of VACOR for this work (Murtagh, 2003), and the results obtained will be discussed in the next section.

6.2 Atypical Direction of Price Movements

Typical movements can be read off in percentage terms in Tables 1-4. More atypical movements serve to define the strong patterns in our data.

We consider the clusters of current time-step code categories numbered 65, 68, 69, 70, 71, 72, 73 from Table 5, as discussed above in section 5, and we ask what are the likely movements, for one time step. Alternatively expressed the current code categories are defined at time step t , and the one-step-ahead code categories are defined at time step $t + 1$. Projections (e.g. Figure 2) are descriptive (“what is?”), but correlations and contributions point to influence (“what causes?”). Correlations and contributions are used therefore, as shown in Table 5, in preference to projections.

We find the following predominant movements in Table 5, using a threshold CTR value of 0.3:

- Cluster 65, i.e. code category 9: \longrightarrow weakly 8 and more weakly 9.
- Cluster 68, i.e. code categories 2 and 3: \longrightarrow 7.
- Cluster 69, i.e. mixed code categories: \longrightarrow 6.
- Cluster 70, i.e. code category 10: \longrightarrow 10.
- Cluster 71, i.e. code category 8: \longrightarrow weakly 8.
- Cluster 72, i.e. code category 1: \longrightarrow 1.
- Cluster 73, i.e. code category 5: \longrightarrow 5

Consider the situation of using these results in an operational setting. From informative structure, we have found that code category 1 (values less than the 10th percentile, i.e. very low) has a tendency, departing from typical tendencies, to be prior to code category 1 (again very low). From any or all of Tables 1–4 we can see how often we are likely to have this situation in practice: 19.04% (= average of 23.29%, 17.67%, 16.4%, 18.8%), given that we have code category 1.

In Table 6 we used clustered one-step-ahead (i.e., output or column) codes. We find the following predominant movements in Table 6:

- Cluster 65, i.e. code category 9: \rightarrow inconclusive.
- Cluster 68, i.e. code categories 2 and 3: \rightarrow 2, 3, 7.
- Cluster 69, i.e. mixed code categories: \rightarrow inconclusive.
- Cluster 70, i.e. code category 10: \rightarrow 8, 9, 10.
- Cluster 71, i.e. code category 8: \rightarrow weakly inconclusive.
- Cluster 72, i.e. code category 1: \rightarrow 1.
- Cluster 73, i.e. code category 5: \rightarrow 4, 5, 6.

From the foregoing, a possible intersection set of clusters derived from the clusters of current, and one-step-ahead future, values is:

1; 2,3; 4,5,6; ignore 7; 8,9,10.

Applying a similar fingerprinting analysis to Ross's (2003) oil data, 749 values, we found that clustering the initial code categories did not make much sense: we retained therefore the trivial partition with all 10 code categories. For the output or one-step-ahead future code categories, we agglomerated 6 and 7, and denoted this cluster as 11. Table 7 shows the results. We find the following, generally weak, associations derived from the contributions (second of the two columns in Table 7: we used approximately 0.3 as the cut-off value).

- Input code category 6 \rightarrow output code categories 1, 10 (weak).
- Input code category 3 \rightarrow output code category 2.
- Input code category 4 \rightarrow output code category 4.
- Input code categories 9, 2 \rightarrow output code category 5 (weak).
- Input code category 10 \rightarrow output code category 8.

Not surprisingly, we find very different patterns in the two data sets of different natures used, the futures and the oil price signals.

7 Conclusions

Correspondence analysis has been shown to be a flexible, robust and scalable environment for data analysis of what presents itself initially as structureless data. Correspondence analysis involves the alternative viewpoints offered by use of three metrics: (i) the χ^2 metric defined on profiles; (ii) the Euclidean metric, defined on factors, and far more conducive to display than the χ^2 metric; and (iii) the ultrametric, associated with a hierarchical clustering or tree representation, and permitting code category aggregation. The latter property of the analysis allows us to search for appropriate resolution level for the analysis.

We have shown that structure can be discovered in data where such structure is not otherwise apparent. Furthermore we have used correspondence analysis, availing of its spatial projection and clustering aspects, as a convenient analysis environment. Validating the conclusions drawn is always most important, and this is facilitated by semi-interactive data analysis.

Acknowledgements

This work was carried out in the context of the project “Integration Methods in Financial Analysis”, led by Dr Patrick Muldowney, University of Ulster, and supported by the British Council, UK, and the Polish State Committee for Scientific Research, KBN, Poland. Discussions with Dr Muldowney are appreciated. Discussions with Dr T.K. Gopalan, Chennai, India, on VACOR and other aspects of correspondence analysis are also appreciated.

References

1. Benzécri, J.P. (1976). *L'Analyse des Données. Tome II. L'Analyse des Correspondances*, 2nd ed., Dunod.
2. Benzécri, J.P. (1992). *Correspondence Analysis Handbook*, Marcel Dekker.
3. Doob, J.L. (1953). *Stochastic Processes*, Wiley.
4. Jambu, M. (1978). *Classification Automatique pour l'Analyse des Données, 1 – Méthodes et Algorithmes*, Dunod, Paris.
5. Mantegna, R.N. & Stanley, H.E. (2000). *An Introduction to Econophysics*, Cambridge University Press.
6. Murtagh, F. (1985). *Multidimensional Clustering Algorithms*, Physica-Verlag.
7. Murtagh, F. & Sarazin, M. (1993). Nowcasting astronomical seeing: a study of ESO La Silla and Paranal. *Publications of the Astronomical Society of the Pacific*, 105, 932–939.
8. Murtagh, F., Aussem, A. & Sarazin, M. (1995). Nowcasting astronomical seeing: towards an operational approach. *Publications of the Astronomical Society of the Pacific*, 107, 702–707.
9. Murtagh, F. (2003). MDA-J: Multivariate Data Analysis – Java, <http://astro.u-strasbg.fr/~fmurtagh/mda-sw>
10. Ross, S.M. (2003). *An Elementary Introduction to Mathematical Finance*, 2nd ed., Cambridge University Press.
11. Samuelson, P.A. (1965). Proof that properly anticipated prices fluctuate randomly. *Industrial Management Review*, 6, 41-45.
12. Hongyuan Zha, Xiaofeng He, Ding, C., Simon, H., & Ming Gu (2001). Bipartite graph partitioning and data clustering. *Proc. ACM 10th International Conference on Information and Knowledge Management – CIKM 2001*, Nov. 5–10, Atlanta, GA, pp. 25–31.

Table 1: Cross-tabulation of log-differenced futures data using quantile coding with 10 current and next step price movements. Values 1 to 2500 in the time series are used. Cross-tabulation results are expressed as percentage (by row).

	j1	j2	j3	j4	j5	j6	j7	j8	j9	j10
i1	23.29	7.23	8.84	6.02	14.86	1.20	10.44	8.84	8.43	10.84
i2	11.60	11.60	11.20	8.80	13.20	5.20	11.60	8.80	8.80	9.20
i3	10.00	13.20	10.80	12.80	14.40	2.00	12.80	5.60	10.80	7.60
i4	8.00	9.20	9.20	12.00	15.60	4.80	12.00	10.40	9.60	9.20
i5	7.50	9.50	9.75	11.00	22.25	5.25	7.50	10.25	9.00	8.00
i6	5.05	8.08	9.09	10.10	20.20	6.06	9.09	16.16	4.04	12.12
i7	4.80	9.60	12.40	11.60	21.60	2.40	10.40	9.20	10.40	7.60
i8	8.40	7.20	8.40	12.40	13.20	7.20	8.40	10.80	11.60	12.40
i9	8.40	12.00	8.40	6.80	15.60	2.00	10.00	13.60	9.60	13.60
i10	11.20	11.60	11.60	8.00	8.00	4.00	8.80	10.00	14.80	12.00

Table 2: Cross-tabulation of log-differenced futures data. Values 3001 to 5500 expressed as percentage (by row).

	j1	j2	j3	j4	j5	j6	j7	j8	j9	j10
k1	17.67	13.65	9.24	7.22	12.05	3.21	10.44	10.84	8.03	7.63
k2	12.40	9.20	11.60	10.00	12.00	4.00	11.60	10.80	10.40	8.00
k3	12.00	11.60	11.60	9.20	18.40	3.60	12.00	6.80	6.40	8.40
k4	10.00	10.80	12.00	9.60	16.40	3.20	10.80	7.20	6.80	13.20
k5	8.78	6.59	11.46	10.98	21.71	4.63	10.24	8.54	9.27	7.80
k6	3.37	11.24	10.11	15.73	19.10	3.37	5.62	7.87	11.24	12.36
k7	7.60	11.60	7.20	9.60	16.40	4.40	9.60	12.40	12.40	8.80
k8	9.60	7.60	9.20	9.60	20.00	3.60	9.60	10.80	9.60	10.40
k9	5.60	10.00	8.80	13.20	12.80	3.20	9.60	14.00	14.00	8.80
k10	9.60	10.40	8.00	8.00	13.60	1.60	7.60	10.40	13.20	17.60

Table 3: Cross-tabulation of log-differenced futures data. Values 2001 to 4500 expressed as percentage (by row).

	j1	j2	j3	j4	j5	j6	j7	j8	j9	j10
m1	16.40	14.40	7.60	9.20	11.20	4.40	11.60	6.40	9.60	9.20
m2	11.64	10.84	12.45	10.04	12.45	4.02	9.64	10.44	10.04	8.43
m3	12.80	13.60	11.60	8.00	15.20	3.20	11.20	9.20	7.60	7.60
m4	10.40	12.80	8.80	10.40	10.00	6.00	12.40	9.60	7.20	12.40
m5	7.68	6.88	9.79	9.26	22.49	6.08	8.99	9.79	11.11	7.94
m6	7.38	8.20	9.02	12.30	22.95	2.46	6.56	9.84	9.84	11.48
m7	7.23	9.24	7.23	12.05	17.27	7.23	8.03	10.44	12.05	9.24
m8	10.80	6.80	10.40	6.80	15.20	4.80	12.40	12.00	8.40	12.40
m9	5.20	9.60	12.80	12.00	13.60	4.80	8.80	12.40	11.60	9.20
m10	10.40	8.00	10.00	11.60	11.20	4.00	8.80	10.00	12.00	14.00

Table 4: Cross-tabulation of log-differenced futures data. Values 3601 to 6100 expressed as percentage (by row).

	j1	j2	j3	j4	j5	j6	j7	j8	j9	j10
n1	18.80	14.40	7.60	8.00	8.80	3.20	10.40	11.20	8.40	9.20
n2	12.40	9.60	11.20	9.60	13.20	3.60	12.00	10.00	10.40	8.00
n3	7.97	11.55	10.76	9.96	19.92	5.18	11.55	9.16	7.97	5.98
n4	9.64	10.04	11.24	10.44	16.87	4.02	7.63	8.84	8.43	12.85
n5	9.65	7.92	12.38	9.16	21.29	4.70	11.63	6.93	9.16	7.18
n6	6.38	5.32	12.77	15.96	18.09	7.45	8.51	6.39	7.45	11.70
n7	7.60	11.60	6.80	8.00	17.60	4.40	10.80	12.00	12.40	8.80
n8	9.16	7.57	9.56	11.55	17.93	3.98	9.96	11.16	7.97	11.16
n9	4.82	10.44	8.84	12.85	15.26	1.61	9.64	12.85	14.06	9.64
n10	11.60	10.00	9.20	8.40	11.20	1.20	6.00	11.60	12.40	18.40

Table 5: Table crossing clusters (on I) and coordinates (J), giving correlations and contributions (as thousandths). Clusters retained here: 65, 68, 69, 70, 71, 72, 73. Coordinates: j1, j2, ... j10.

Top of hierarchy agglomerations:

((65 (73 (69 71))) (70 (68 72)))

Cluster 65: k9 n9 k7 n7 i4 m9	Predominant: 9
Cluster 68: i3 k3 m3 m4 i2 m2 k2 n2	Predominant: 2, 3
Cluster 69: n6 i8 m7	Predominant: none
Cluster 70: i10 m10 i9 k10 n10	Predominant: 10
Cluster 71: i6 k4 n4 m8 k8 n8	Predominant: 8
Cluster 72: i1 m1 k1 n1	Predominant: 1
Cluster 73: i5 m5 n3 k5 n5 k6 i7 m6	Predominant: 5

Clusters 65 through 73 represent the input coding categories.

Coordinates j1 through j10 represent the output coding categories.

	j1		j2		j3		j4		j5		j6		j7		j8		j9		j10	
	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR
65	432	131	6	10	40	180	42	76	9	5	3	3	2	5	190	452	225	308	52	43
68	187	52	129	210	77	315	7	11	170	86	1	1	176	470	76	166	50	64	128	99
69	114	36	138	254	5	22	189	353	1	1	478	665	44	133	14	34	4	5	13	11
70	3	2	3	8	6	40	52	150	188	167	105	226	52	243	21	79	122	269	449	607
71	40	7	153	157	5	13	17	18	154	49	10	8	0	0	56	78	368	293	196	96
72	728	661	48	256	23	308	55	297	89	147	22	88	6	54	5	35	14	58	9	22
73	210	112	34	104	16	122	31	96	567	545	4	9	19	95	37	155	1	2	83	121

Table 6: Table crossing clusters on I with clustered output one-step-ahead code categories j1, j14, j15, j16. Correlations and contributions are given as thousandths.

Top of hierarchy agglomerations:
 ((65 (73 (69 71))) (70 (68 72)))

Cluster 65: k9 n9 k7 n7 i4 m9	Predominant: 9
Cluster 68: i3 k3 m3 m4 i2 m2 k2 n2	Predominant: 2, 3
Cluster 69: n6 i8 m7	Predominant: none
Cluster 70: i10 m10 i9 k10 n10	Predominant: 10
Cluster 71: i6 k4 n4 m8 k8 n8	Predominant: 8
Cluster 72: i1 m1 k1 n1	Predominant: 1
Cluster 73: i5 m5 n3 k5 n5 k6 i7 m6	Predominant: 5

Clusters 65 through 73 represent the input coding categories.
 Coordinates j1 through j10 represent the output coding categories.

Top of hierarchy agglomerations for output coding categories:
 (1 (16 (14 15)))

	j1	j14	j15	j16
	j1	j2,j3,j7	j8,j9,j10	j4,j5,j6
	very low	low, spoiled	high/ very high	middle
	COR CTR	COR CTR	COR CTR	COR CTR
65	734 131	4 4	260 114	2 0
68	201 52	399 583	264 169	137 52
69	210 36	261 250	2 1	527 132
70	4 2	26 57	568 543	402 229
71	299 7	239 32	17 1	445 15
72	784 661	8 37	29 60	179 221
73	277 112	17 38	114 113	592 349

Table 7: Table crossing I with a partition of J . Correlation and contributions shown. Ross (2003) oil data. Values given as thousandths.

	1		2		3		4		5		8		9		10		6,7	
	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR	COR	CTR
1	208	73	7	7	225	99	85	36	239	107	9	4	9	6	211	85	7	2
2	150	143	73	208	72	86	31	36	249	303	3	4	152	269	78	85	193	138
3	47	23	287	420	335	207	42	25	14	8	163	110	59	54	49	28	4	1
4	4	4	23	86	107	166	317	473	64	101	2	3	122	281	64	91	296	274
5	221	125	0	0	152	108	115	79	71	51	139	108	126	132	137	90	38	16
6	228	295	10	39	126	204	126	197	34	57	11	19	61	148	48	71	356	345
7	347	69	126	74	105	26	29	7	2	0	154	41	147	54	11	2	79	12
8	46	26	48	82	52	37	52	36	73	53	249	195	4	5	414	274	62	27
9	5	3	6	10	55	37	43	28	460	319	48	35	5	5	6	4	373	152
10	235	237	24	73	23	29	68	83	0	0	350	481	25	46	231	268	44	33

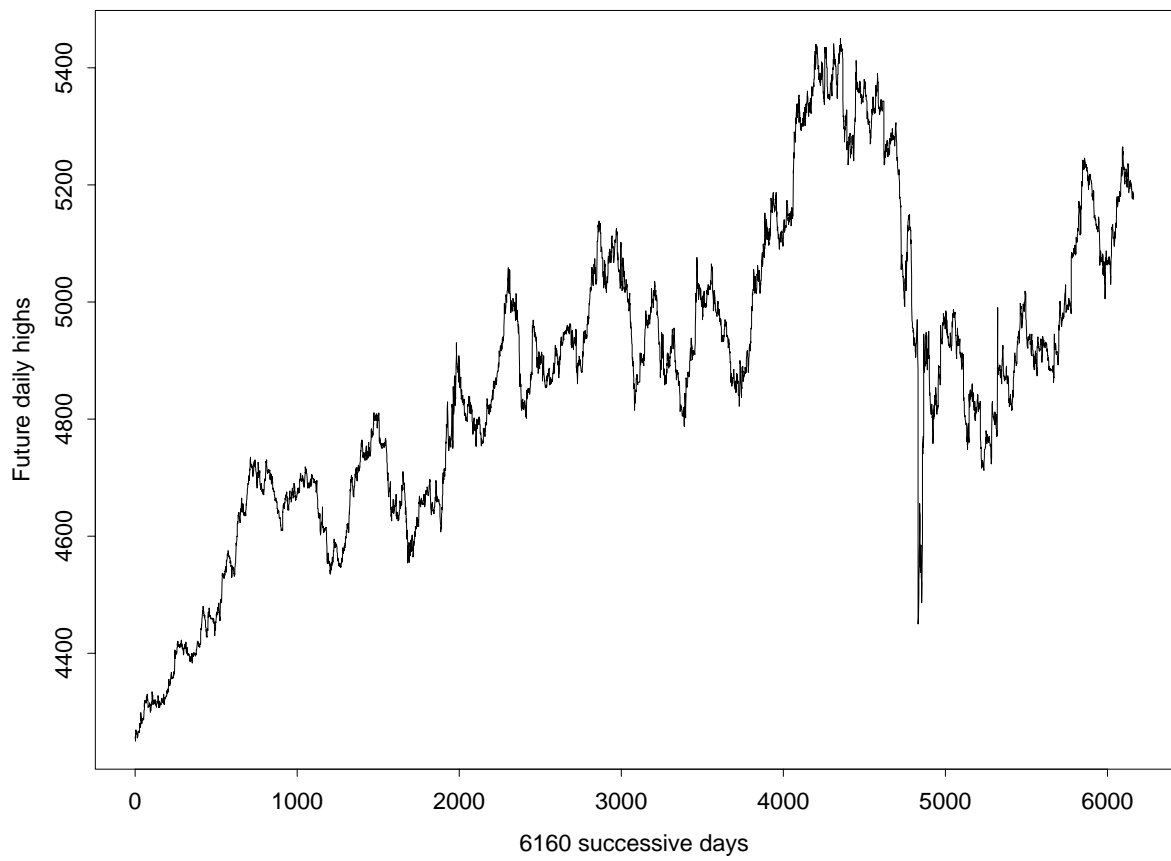


Figure 1: Future daily highs.

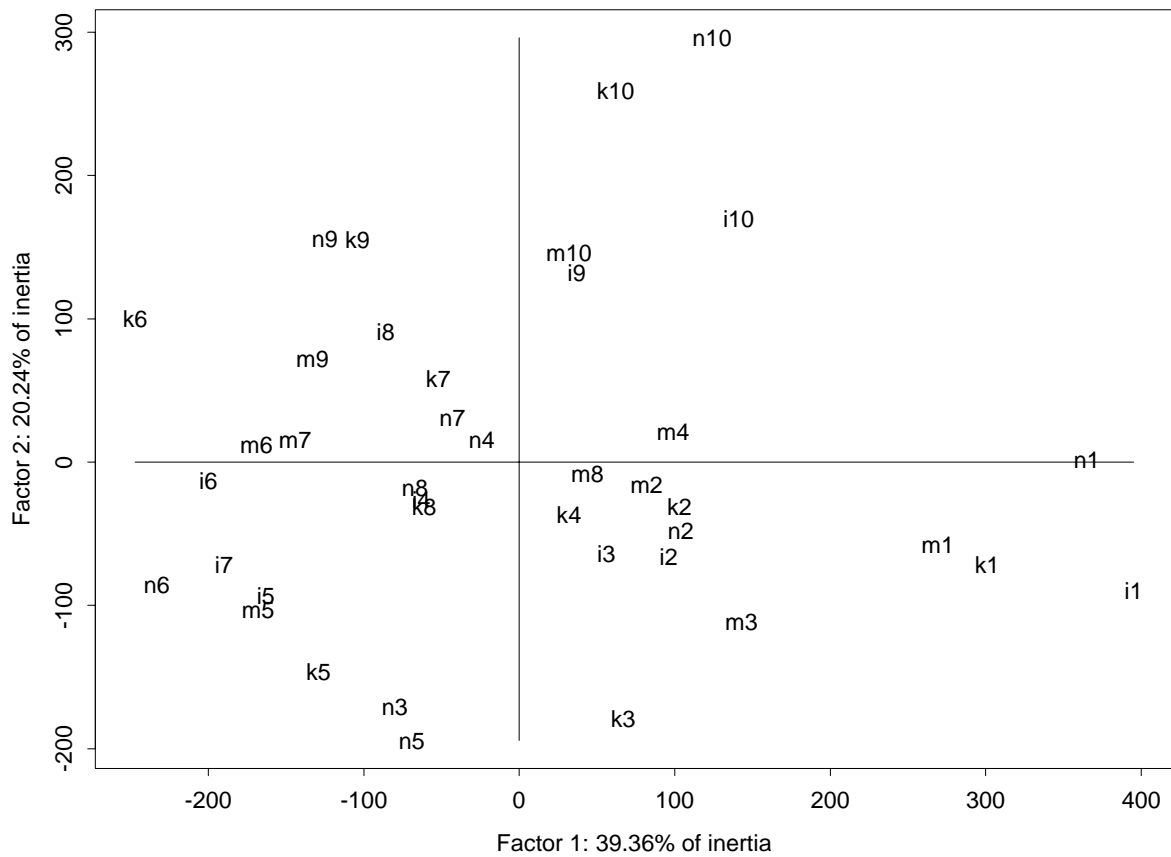


Figure 2: Factors 1 and 2 with input code categories 1 through 10 defined on 4 different spanning segments of the input data signal. Only input, or current, values are displayed here. The 4 time series sub-intervals are represented by (in sequential order) *i*, *m*, *k*, *n*. The quantile coding is carried out independently in each set of 10 categories.

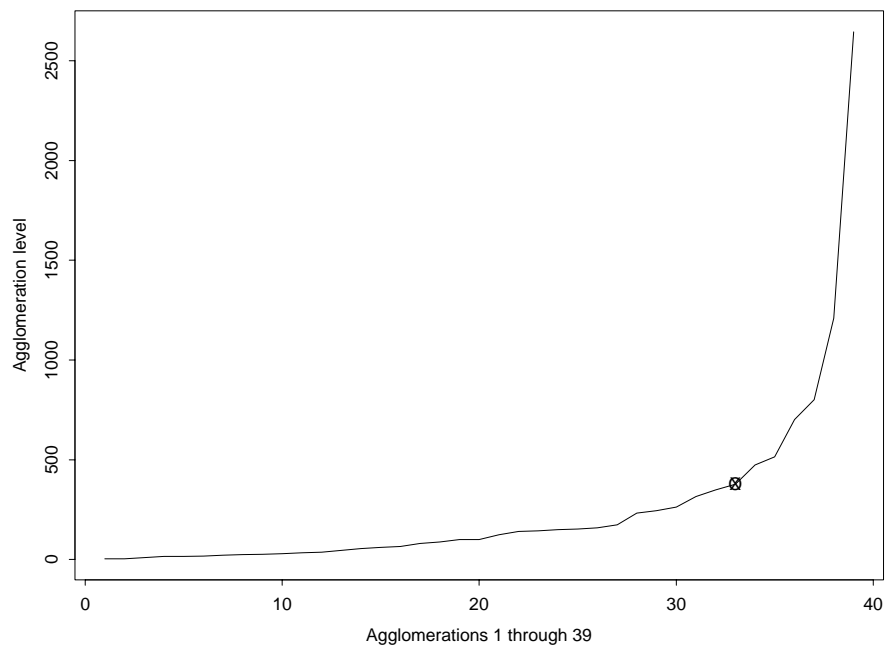


Figure 3: Cluster agglomeration levels for the hierarchical clustering (minimum variance criterion, factor projections used as input) of the 40 observations i, k, m, n in Tables 1–4.

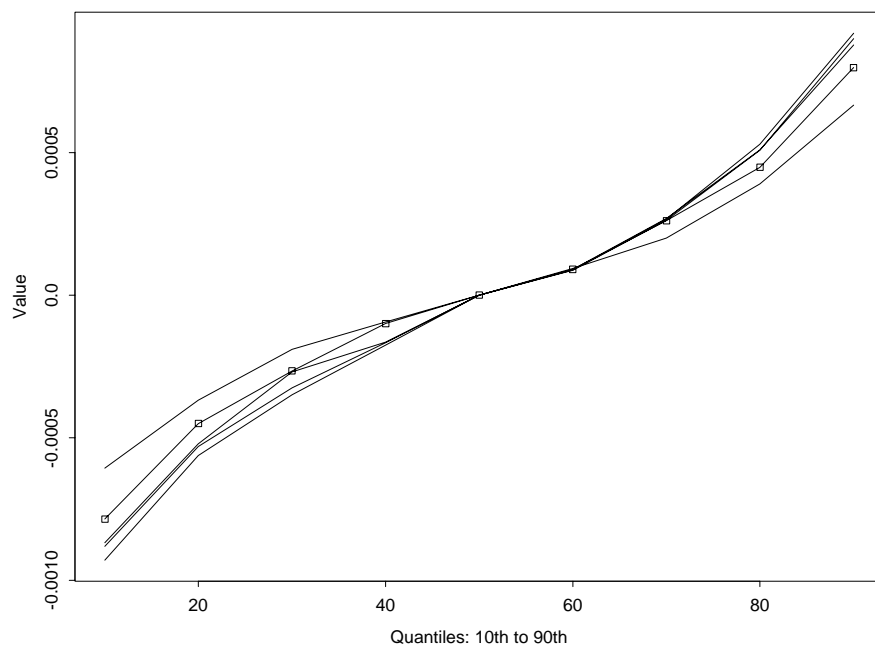


Figure 4: Stationarity of coding interval boundaries: Quantile values derived from Tables 1-4, and – box points – averaged (by quantile).