Appendix

A.1. Text Preprocessing for Sentiment and Document Term Matrix Extraction

Cleaning the data involves (i) converting all text into lowercase, (ii) removing stop words (e.g. *it*, *out*, *so*, and *the*) though not negating stop words (like *no*, *nor*, and *not*), (iii) tokenizing the text (e.g. converting *Boston-based* to *Boston* and *Based* as separate words), (iv) entity replacement (for example, *International Monetary Fund* → *IMF*, numbers → tokens *_n_*, *_mn_*, and *_bn_*), (v) sentiment negation using the Das and Chen (2007) algorithm, (vi) punctuation removal, and (vii) word stemming (which converts inflected words, like *cats* or *speaking* into their root form).

Once data have been cleaned, we select all relevant English language articles for either the EM or DM corpus. Articles in the Thomson Reuters archive are often revised several times after their initial publication. Such article chains, i.e. the initial article and subsequent revisions, are labeled with a unique Primary News Access Code (PNAC) code. For each PNAC code, we select only the final article in the sequence. If we were more focused on a high frequency analysis, it would be more natural to select the first rather than last article, but for the time horizon of our analysis (monthly) we believe that the final article is likely to have the richest information content and the several hour lag from first to last article in a chain will not have a meaningful effect on our results.

A.2. Construction of Econ Word List

The initial list of economics words (which we refer to as *econ words*) was compiled by the authors by looking at every word in the index of Beim and Calomiris (2001), and then

subjectively selecting words with important economic or market-related meaning. This yielded 237 words. Then using the articles from the Thomson-Reuters corpus from 1996-2015 that were tagged by the publisher as being about emerging markets (having a *qcode* of *N2:EMRG*), we analyzed all words occurring more than 3,000 times in any given year. This yielded 3,831 unique words. We ranked these words by their *cosine similarity* (see definition in the next section) to our original set of 237 words, averaged over all years in which these words appeared more than 3,000 times. Out of those words with an average cosine distance greater than 0.015 (which can be thought of as roughly a correlation of 1.5%), the authors and their research assistants selected an additional set of words that co-occurred very frequently with the original set of words. We then culled our list to eliminate redundancy (words that have a common word stem).

We then added 59 more commodity-related words by looking up the commodity groups from the IMF's *Indices of Primary Commodity Prices*[29], 18 subjectively determined housing-related words, and 8 law-related words.

As a final step for identifying econ words, we collected the most frequently occurring 500 bigrams in every year of our Thomson-Reuters emerging markets article set. This yielded 2,052 unique bigrams (for example, the two most frequently occurring bigrams were "Reuters message", which we deemed not useful, and "central bank'", which we deemed useful) and for the 100 bigrams which we subjectively determined to be economically relevant, we replaced the bigram with a single token which would then appear in our document term matrixes (and therefore in our topic analysis). For example, the bigram "central bank" was replaced with the

[29] See https://www.imf.org/external/np/res/commod/Table1a.pdf.

token *central_bank*. We repeated the same analysis for the top 500 trigrams in every year. These yielded 3,740 unique trigrams, of which we determined 13 to be relevant (for example, the most frequently occurring retained trigram was "International Monetary Fund"). There were many fewer retained trigrams than bigrams because having a three-word phrase introduces a much greater degree of context than does a 2-word phrase, which renders many of the examined trigrams not broadly useful.

This process yields a total of 1,242 unique tokens (words and tokenized bigrams and trigrams) for constructing our document term matrixes.

A.3. Topic Extraction using the Document Term Matrix

We consider two words to be closely connected – or to co-occur – if there are many articles in which the two words appear together. Our measure of co-occurrence is the *cosine similarity* between two words. This similarity measure is computed as $\frac{D_j' D_i}{\|D_i\|\|D_j\|}$ where D_i is the i^{th} column of the document term matrix, and $\|D_i\|$ is the Euclidean norm of the i^{th} column. The cosine similarity has several nice properties: it is zero for words that never occur together in the same document, it is 1 for a word relative to itself, and it is larger for words that, conditional on how often they occur, tend to occur in articles together. Let us refer to the symmetric matrix whose element $A_{i,j}$ corresponds to the cosine similarity between words *i* and *j* as the co-occurrence matrix. The matrix *A* defines a network of our econ words, where the strength of the link between two words corresponds to their cosine similarity.

We are now interested in extracting the structure of this network by finding non-overlapping clusters of words (i.e. a words appears in only one cluster) that tend to occur

together frequently. We will refer to such word clusters as *topics*. Here we follow the approach of Newman and Girvan (2004) and Newman (2006), and cluster our word network so as to maximize network modularity – which we do via the Louvain algorithm (see Blondel et al. 2008). For a given partition of a network into k communities, let us define the $k \times k$ symmetric matrix e as having its $(i,j)^{th}$ element equal to the fraction of all edge weights in the network that connect members of communities i and j. The modularity of the network, $Q = \text{Trace } e - |e^2|$, where $|\cdot|$ indicates the sum of matrix elements, is a measure of the extent to which intra-cluster links tend to occur more frequently than at random. The Louvain algorithm is a particularly effective maximization heuristic for finding network partitions to maximize modularity.

Figure 3 shows the initial clustering produced by the Louvain method for our EM and DM co-occurrence network. Clusters are ordered from the largest (by number of words) to the smallest, with the number of words in a cluster on the y-axis. As is evident, the algorithm naturally produces five large clusters for the EM and DM corpora, as well as a collection of several dozen much smaller clusters. Following the initial Louvain clustering, we then place each word from a small cluster (i.e. one outside of the top five) into one of the top five clusters so as to maximize network modularity. This process then yields five EM and DM clusters – each of which is a subset of our 1,242 econ words.

To investigate the time stability of our clustering algorithm, we repeated our topic extraction over successive 4 year windows of our DM and EM corpora (recall the data set runs from 1996 to 2015). In each 4-year window we recalculated the modularity maximizing word clusters using the Louvain method. To compare the subsample word categories to the full sample ones, we use the *best match* method described in Section 2.2 of Meila (2007). Consider two sets of word topics, C and C', defined over the same set of words. For each topic k in C we

find the topic *k'* in *C'* which has the maximum word overlap with *k*, while making sure that the mapping is injective – i.e. that a topic *k'* in *C'* only gets mapped into once (if at all, because *C* and *C'* may not have the same number of topics). We then count the total number of words in the best-matched topics in *C* and *C'*, and divide this by the total number of econ words appearing in both clusterings. This measure tells us what fraction of all our econ words fall into the same topic category in *C* and *C'*, where "same" means the best-matched categories.

In the five 4-year subsamples of our DM corpus, we find that the fractions of words matched from each subsample set of categories to the same full sample ones are 70% (1996-1999), 79% (2000-2003), 80% (2004-2007), 84% (2008-2011) and 78% (2012-2015) respectively. So approximately 80% of all our econ words get placed into the same topic in the full sample and in each of our subsamples. For our EM corpus, these fractions are 72%, 77%, 74%, 78% and 67%. In the last 4 years (2012—2015) of our EM sample, the full sample *Mkt* topic is split between the subsample *Macro* and *Mkt* topics, and the full sample *Comms* topic is split between the subsample *Comms* and *Mkt* topics; these account for the somewhat lower word overlap in this sub-period.

It should be noted that under the null that the full sample clustering is identical to each subsample, we still wouldn't expect to empirically find 100% cluster overlap due to sampling variation. Furthermore, we do not weigh overlapping words by frequency of occurrence, and if we were to do so, we would find higher percent overlap than the reported numbers. We interpret these results as indicating that the topics we identify over the full sample are quite robust, and a subsample-level analysis identifies very similar topics to the full sample. In the paper, we present all our results using our full sample topics.

We also investigated the potential usefulness of an alternative method – Latent Dirichlet Allocation (LDA) – for defining topic areas. In LDA, words are not assigned to mutually exclusive groups, but rather a group is defined as a probability distribution over all the words. We performed a pilot study to investigate the sensitivity of our results to the use of the Louvain method as opposed to LDA. We found no major qualitative differences in the resultant topics from the two methods. However, the Louvain method is much faster. For example, the document term matrix for our EM sample has 4,994,729 rows and 1,240 columns (our EM sample has no occurrences of 2 of our econ words). The computation time for the Louvain method, which involves computing cosine similarity for all word pairs and finding clusters to maximize network modularity, is 40 seconds. Computing the LDA clustering with 5 topics for *only a single month* (February 2007) took 113.8 seconds, and computing LDA for all of 2007 took 1,708 seconds. Assuming LDA scales roughly linearly, it would take approximately 10 hours for our entire sample. We, therefore, decided to focus on the Louvain method, given its relative ease of computation.

A.4. n-grams and Conditional Probabilities

The count operators $\hat{c}$ in equation (1) return the number of occurrences of a given 4-gram w_1, w_2, w_3, w_4 and its starting 3-gram w_1, w_2, w_3 in the training corpus. For month *t*, the training corpus for our EM (DM) entropy measure contains all EM (DM) articles in our sample in the two-year period from month *t*-27 to month *t*-4. We use a rolling two-year window to keep the size of the information set for our entropy calculations constant at all months in our sample. The reason we skip months *t*-3, *t*-2, and *t*-1 is to treat a 4-gram and its starting 3-gram which both

appeared for the first time in month *t*-3, and neither of which had appeared in our corpus before, as being unusual for the next 3 months (such a 4-gram would be assigned *m*=0.1 in months *t*-3, *t*-2, *t*-1, and *t*). In month *t*+1, this 4-gram would be assigned a much higher value of *m* because month *t*-3 would now have entered the training corpus.

We refer to the 1 and 10 present in equation (1) for *m* as the 1:10 rule. Continuing with the example from the prior paragraph, when we observe a 4-gram and its starting 3-gram for the first time we need a rule for assigning an appropriate conditional probability. We would like to treat a new never-before-seen-n-gram as being a representative member of the set of never-before-seen-n-grams. For every month *t*, we find in the EM corpus all 4-grams which do not appear in month *t*'s training corpus. We then compute the *m* for each never-before-seen-n-gram for the remainder of our sample, i.e. from month *t* until December 2015. Tabulating these *m*'s for never-before-seen-n-grams across all months (we have close to 100 million observations) produces the distribution shown in Figure A1 in the Supplementary Appendix. The median value of this distribution is 0.083 and the mean value is 0.28. Therefore, our choice of 1:10 rule would assign an *m* roughly equal to this median to a 4-gram that is encountered in month *t* but not in month *t*'s training corpus.

We experimented using n-grams that drop stop words, and using n-grams that retain stop words. While the results were similar using the two methods, we chose to use n-grams that retain stop words because often these preserve more of the article's semantics. For example, the phrase "business sentiment has improved of late" yields the 4-grams "business sentiment has improved", "sentiment has improved of", and "has improved of late" when stop words are retained. With stop words removed we have "business sentiment improved late" which may convey a different meaning that the original statement.

A.5. Market Price and Macro Data

Our price data come from Bloomberg. Table 1 shows the mapping from each country, as well as for the MSCI EM and DM index, to its corresponding Bloomberg ticker. All stock price data are converted into US dollar terms using end of day exchange rates. Price data are converted into total returns, *return*, by adding in the dividend yield from the prior 12 months accrued over the horizon of the return calculation (either weekly or daily). Our realized volatility variable, *sigma*, is computed by Bloomberg over the last 20 business days of every month. Our drawdown measure, *drawdown*, is computed as the maximum negative return realized by an investment in a given market index over the ensuing 252 trading days. For a given return *return*, we define *retmi* (*retpl*) as $\max(-return, 0)$ ($\max(return, 0)$), i.e. the absolute value of the negative (positive) portion of returns.

To maximize the number of observations for which we have data, we construct our *value* variable to be an accounting-free measure of country-level stock valuation. We borrow ideas from Asness et al. (2013) and define *value* for a country in a month *t* as the average level of the US dollar price of the country's stock market index from 5.5 years to 4.5 years prior to *t*, divided by month *t*'s closing US dollar price of the country index. We obtain similar results in panel regressions where we use the market to book ratio as the value measure, but in this case we lose many observations from our sample because of the lack of book equity data.

Below we document the methodology and data sources underlying our macro data.

- **Rate of growth of real GDP (***gdp*): Quarterly real GDP growth rate data is obtained from International Financial Statistics of the International Monetary Fund. Annual data is used only when quarterly data is not available. The series is calculated as year-over-year percent changes.

- **Rate of growth of GDP deflator** (*gdp_deflator*): Quarterly GDP deflator data is obtained from International Financial Statistics of the International Monetary Fund. Annual data is used only when quarterly data is not available. The series is calculated as year-over-year percent changes.
- **Credit to GDP ratio** (*cp*): We look at domestic credit to private sector as a percentage of GDP. Annual credit to GDP ratio data is obtained from World Development Indicators, World Bank. We use linear interpolation and Bank of International Settlements credit data to replace missing values.
- **First difference of credit to GDP ratio** (*dcp*): First difference of *cp* at the monthly frequency.
- **Interest rate** (*rate*): For developing markets, we use monthly deposit rates from International Financial Statistics of the International Monetary Fund. Deposit rates refer to the weighted average rate offered by commercial and universal banks on three- to six-month time deposits in the national currency. For developed markets, we use government bond yield data from Datastream. The maturity of these yields ranges from 5 to 10 years, with 7 years being the average. We use quarterly data only when monthly data does not exist.
- **Monthly percent change in local currency exchange rate vs the US Dollar** (*dexch*): We obtain the monthly exchange rate data from Datastream. All exchange rates are determined as the US dollar in terms of the local currency (for example, for Turkey, our exchange rate measure is 3.4537 on 12/8/2016). So a positive value of *dexch* represents a local currency depreciation. The US series is set to zero. This variable is truncated at $\pm 50\%$.

- **Pre-election dummy** (*pre*): The pre-election dummy takes a value of one for all months in a six-month window prior to an election, and a value of zero on the election month and all other months. We use the Database of Political Institutions for elections dates.
- **Post-election dummy** (*post*): The post-election dummy takes a value of one for all months in a six-month window after an election, and a value of zero on the election month and on all over months. Any month that would receive a classification as both a pre- and post-election month is labeled as a pre-election (but not a post-election) month. We use the Database of Political Institutions for elections dates.

A.6. Panel Regressions

All panel regressions report robust standard errors using the White method. For the *return* and *sigma* panels we cluster residuals by time to control for cross-sectional correlations between countries. In the $return^{12}$ and *drawdown* regressions, we use Thompson (2011) to cluster by both time and country to control for the serial correlation in our overlapping left-hand-side variables, as well as for country correlations. We use the *plm* package from R for our panel data analysis.

Panel fixed effect regressions, with *N* individuals and *T* time observations, that include lagged, persistent independent variables as regressors (i.e., our one-month ahead volatility regressions in Tables 9 and 10) suffer from a bias in the AR coefficient estimates when *N* is large and *T* is small. Nickell (1981) shows that this bias in the AR(1) case is approximately equal to $-(1+\rho)/(T-1)$ where ρ is the AR(1) coefficient. Since our T is quite large (as large as 200), this bias, which only affects the lagged loadings on one-month realized volatility in one set of panels, is quite small.

Another problem exists with forecasting regressions that use lagged explanatory variables (like price ratios or interest rates) whose changes are correlated with return innovations (see Stambaugh 1999 or Hjalmarsson 2010 for an analysis in a multi-country setting). In this case the coefficient estimate for the explanatory variable, while consistent, will be biased in small samples (in a panel setting Hjalmarsson 2010 points out the bias will be of "second-order"). In our setting this issue may affect the interpretation of the coefficient loadings on our *rate* and *value* measures in our forecasting panels. Ang and Bekaert (2007) use a Monte Carlo study to show that the biases in these coefficient estimates cause them to *underestimate* the effects of rates and dividend yields on future country level returns, and furthermore that the bias is quite small when T is 200 (as is the case here), especially when the forecasting period is a year or longer (as is the case for our 12-month ahead return and drawdown regressions). For these reasons we do not believe that the Stambaugh bias is an important consideration in our setting.

Most importantly, the focus of our analysis in Tables 7-12 and A3-A4 is on the loadings on our sentiment measures. Because these measures are not mechanically related to past returns or drawdowns in the same way that price ratios and interest rates are, there is no reason to be concerned about bias in coefficients estimates for our text measures.

Finally, we investigate whether the non-normality of our *drawdown* variable leads to incorrect inferences in the *drawdown* panels. The residuals from the *drawdown* panels, while not normal, appear quite symmetrical and are as close to normality as the residuals from our $return^{12}$ regressions. We conclude that non-normality is unlikely to be problematic for our estimates.