Datathon Writeup

Team 1

April 21, 2017

1 Introduction

Since the Great Recession, the economy has been steadily improving. One way to track this improvement is by examining the growth in job postings, per month. In this report we focus on the following broad question:

Q: *What is the overall character of the growth in job creation (as measured by postings), and what are the specific forces that drive this growth?*

Our analysis is split into two sections. In the first section, we construct various time-series models with number of job postings as a response. In the second section, we examine job growth by job-category, and examine which *city*-specific factors influence job growth across categories. We give fuller descriptions of these efforts below.

2 A Time-Series Approach to Job Growth

2.1 Understanding and Forecasting Number of Job Openings by Time Series Analysis

In this subsection we conduct thorough, self-content time series analysis for the monthly sampled number of job openings data. The main discoveries are summarized as follows:

1. The raw data series of the number of job openings, which has obvious increasing post financial crisis trend, is highly correlated with other major economic and financial indexes, including unemployment rate, federal fund rates and CPI.

2. After detrending by linear regression fitting, the detrended number of job openings pass the stationarity/unit root tests, and exhibits much smaller correlation with other major indexes. Such a fact justifies the use of stationary ARMA model to fit the detrended job openings series alone, without worrying too much the effects from other indexes.

3. We use both AkaiKe Information Criterion (AIC) and Bayesian Information Criterion (BIC) to select the appropriate order for ARMA model, and it turns out that ARMA(1,1) is the best one. We also show that such the result is robust to the error distribution.

4. To further confirm that univariate model suffices to model the intrinsic relations of the number of job openings, we select a Vector Autoregressive (4), among a couple of candidates multivariate models. We use out-of-sample backtesing to compare the best ARMA(1,1) obtained above and the VAR(4) model, the result shows that the ARMA(1,1) still wins.

5. We conclude that to model the post-financial-crisis number of job openings from a time series point of view, the best model is a time trending, obtained from linear regression against time, plus a ARMA(1,1) stationary error part. Such a model can be in turn used to forecasting the number of job openings in future months.

2.1.1 Data Visualization and Diagnostics

The job openings data analyzed in this subsection is the post financial crisis monthly number of job openings across the country. The time range is from 08/2007 to 01/2017.

In Figure 1 we visualize the time series of the monthly number of job openings, together with other major economic and financial indexes including unemployment rate, interest rate which we chose federal fund rates and CPI. We note that to facilitate the comparison across the four series, we have scaled the four series.

Figure 1: Level of Four Series, Monthly Data from 08/2007 to 01/2017.

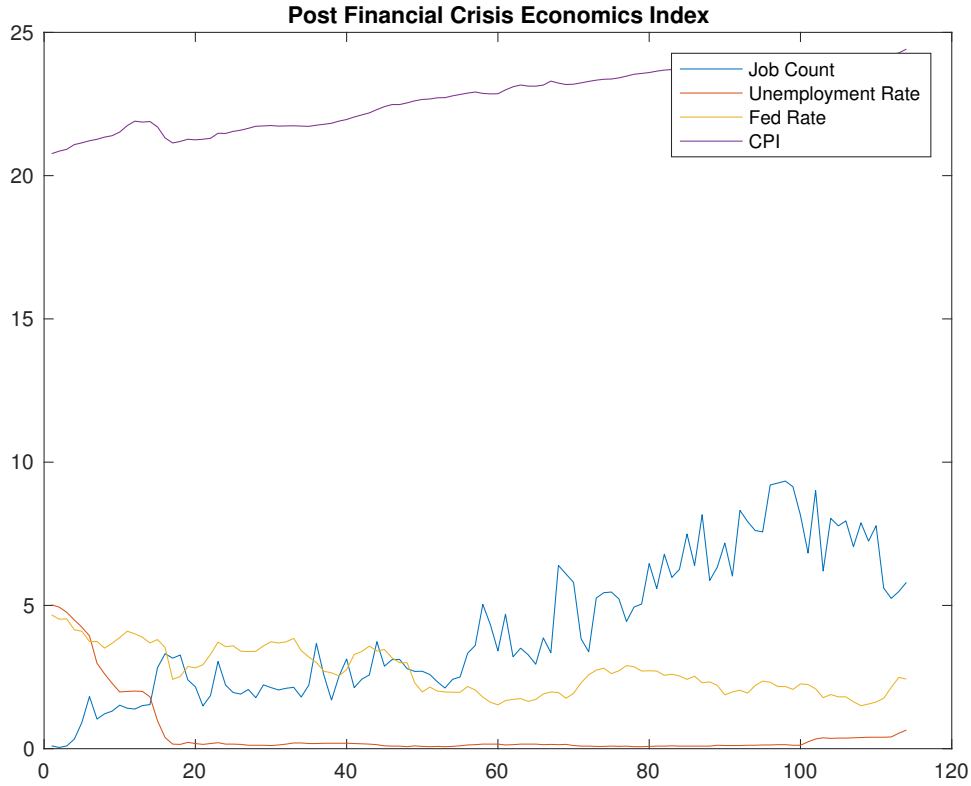
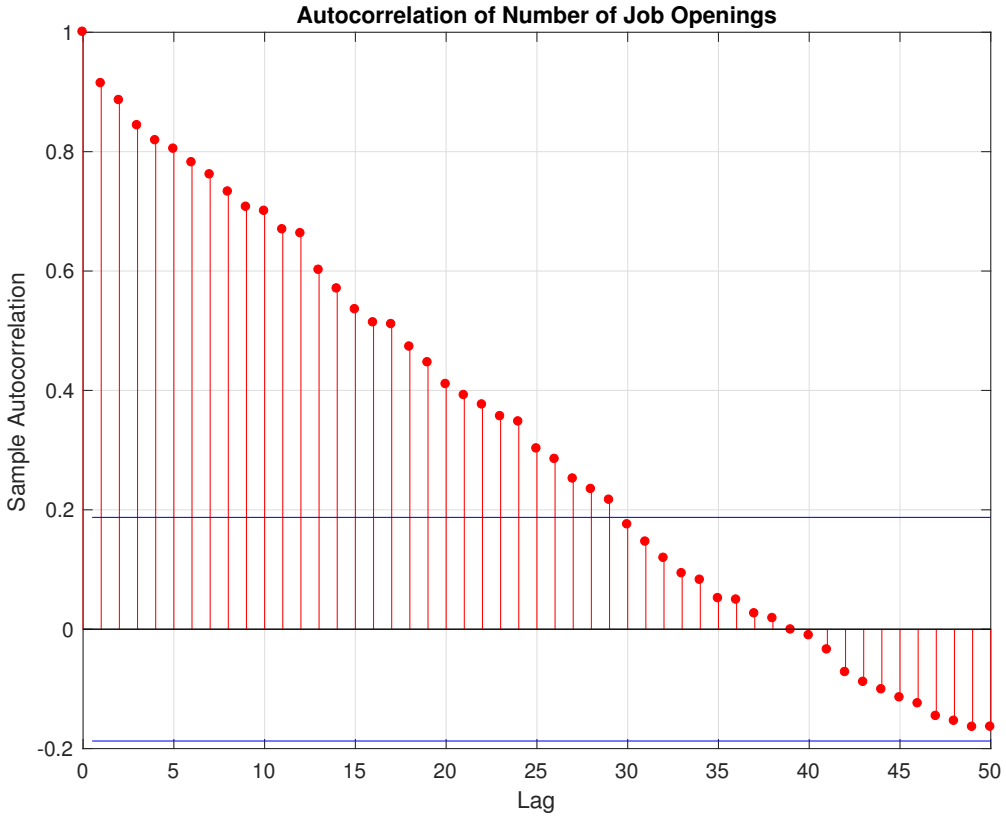
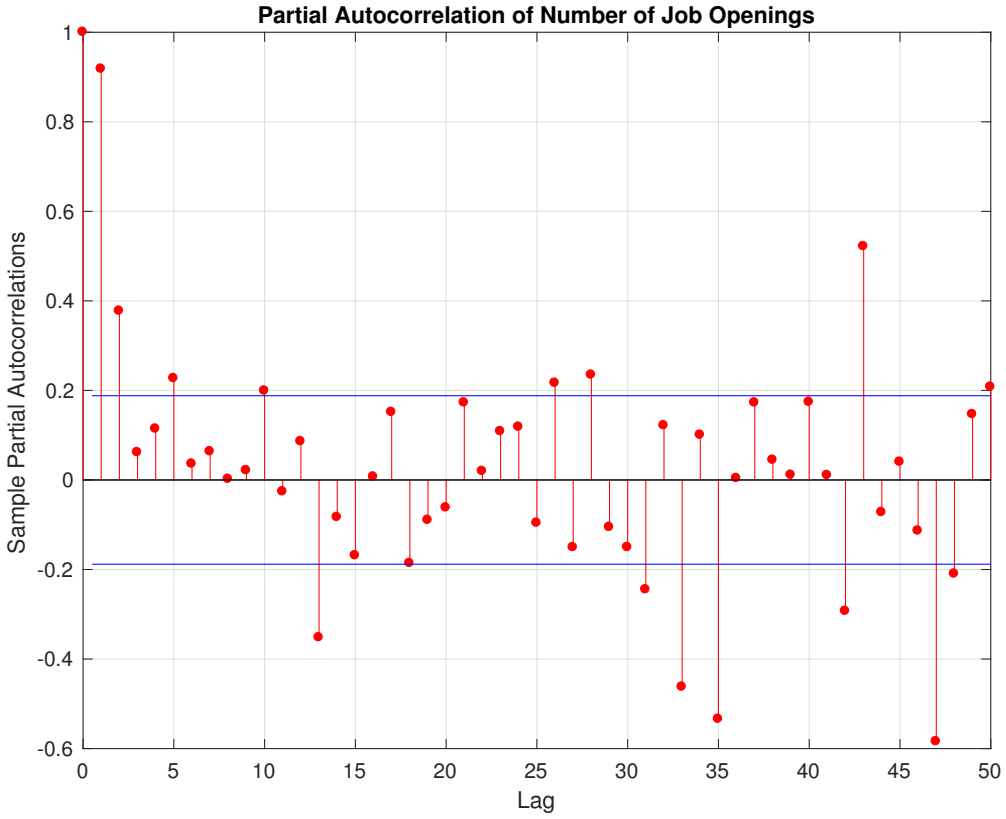
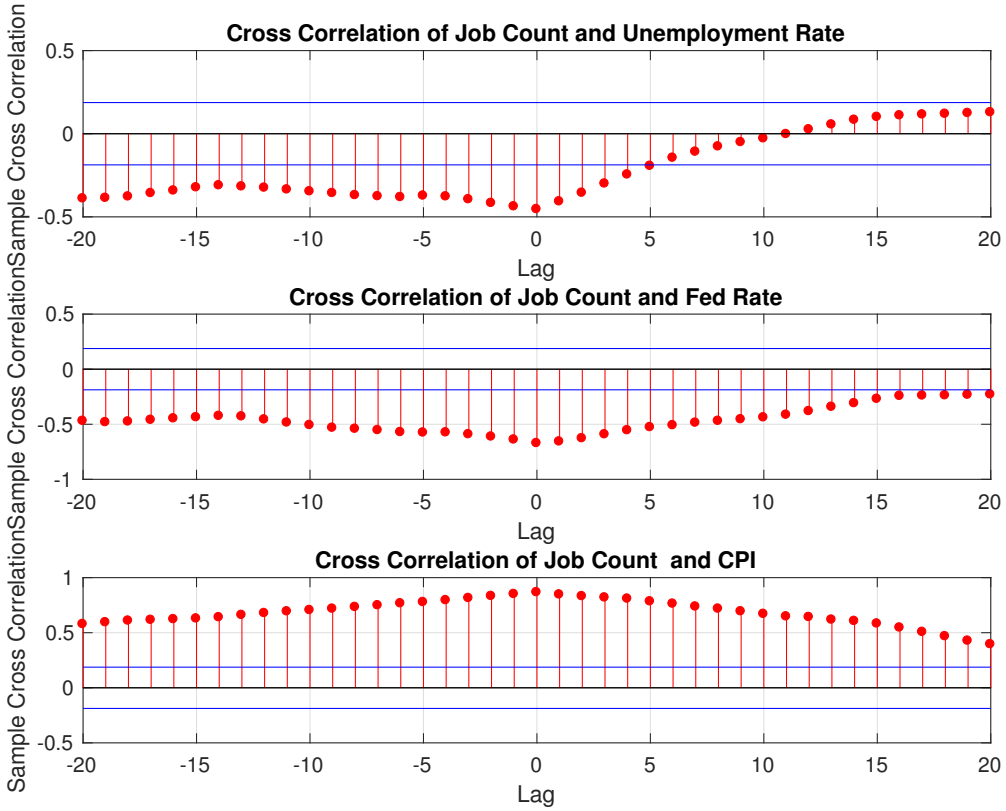


Figure 1 exhibits a clear increasing trending of the number of job openings after the Great Recession. The trending, and hence the induced serial correlation is further confirmed with the autocorrelation Figure 2.1.1 and partial autocorrelation Figure 2.1.1 plots below.


Autocorrelation of Number of Job Openings
Sample Autocorrelation
Lag
1
0.8
0.6
0.4
0.2
0
-0.2
0
5
10
15
20
25
30
35
40
45
50



Moreover, one can observe that such a raw series is positively correlated with CPI and (seemingly) negatively correlated with unemployment rate and fed rate. Such observation can be seen more rigorously from the cross correlation plot Figure 2.1.1.



2.2 Detrending and further Diagnostics

From modelling point of view, trending series is unfavorable as the trending part is always dominating the stationary/stochastic part. In view of this, we first detrend the series and obtained the detrended series. More specifically, assuming, y_t is the number of job openings at month t, we follow the regression

$$y_t = \beta_0 + \beta_1 t + u_t,$$

which yields the Ordinary Least Squared (OLS) estimator for (β_0, β_1) is given by

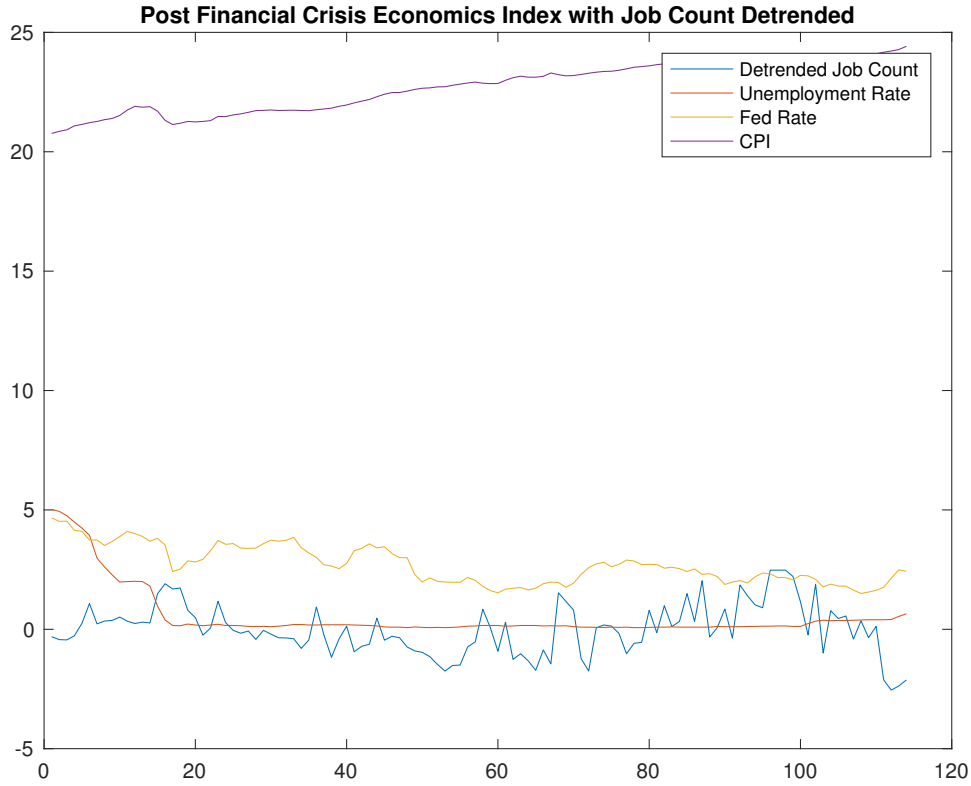
$$\hat{\beta} = (X'X)^{-1}X'Y = (0.3409, 0.0665)'.$$

where X is a matrix of two columns (ones and number of months). Then the detrended series u_t is given by

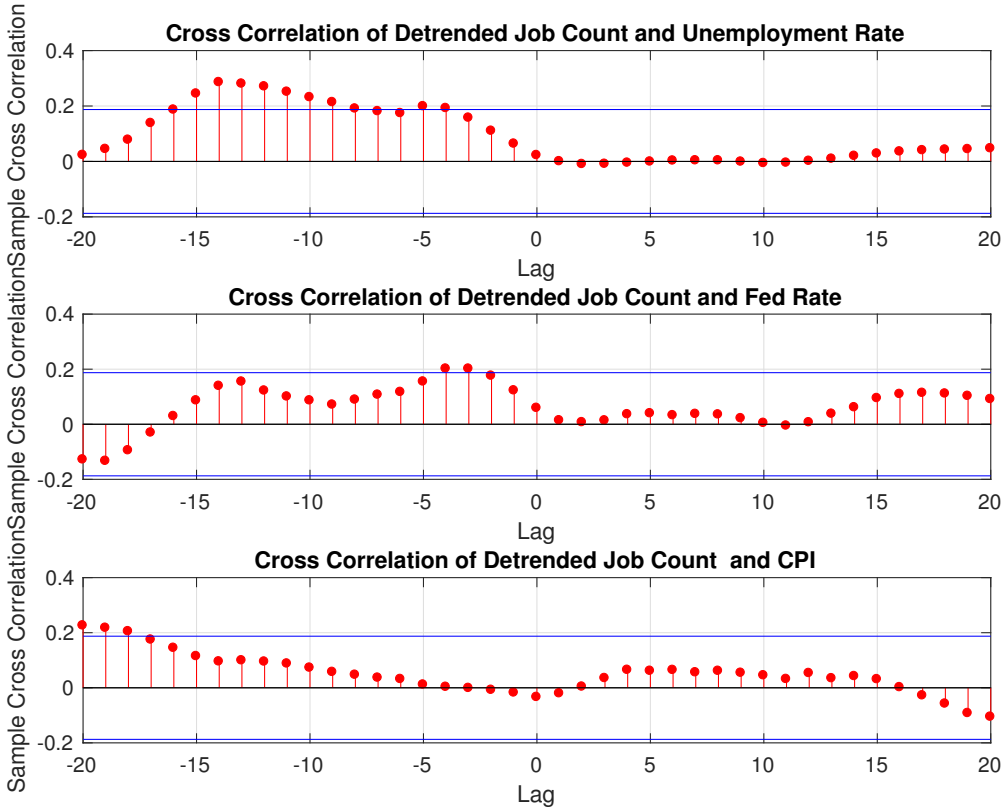
$$\hat{u}_t = y_t - \hat{\beta}_0 + \hat{\beta}_1 t.$$

The resulting detrended u_t, along with other series is visualized in Figure 2.

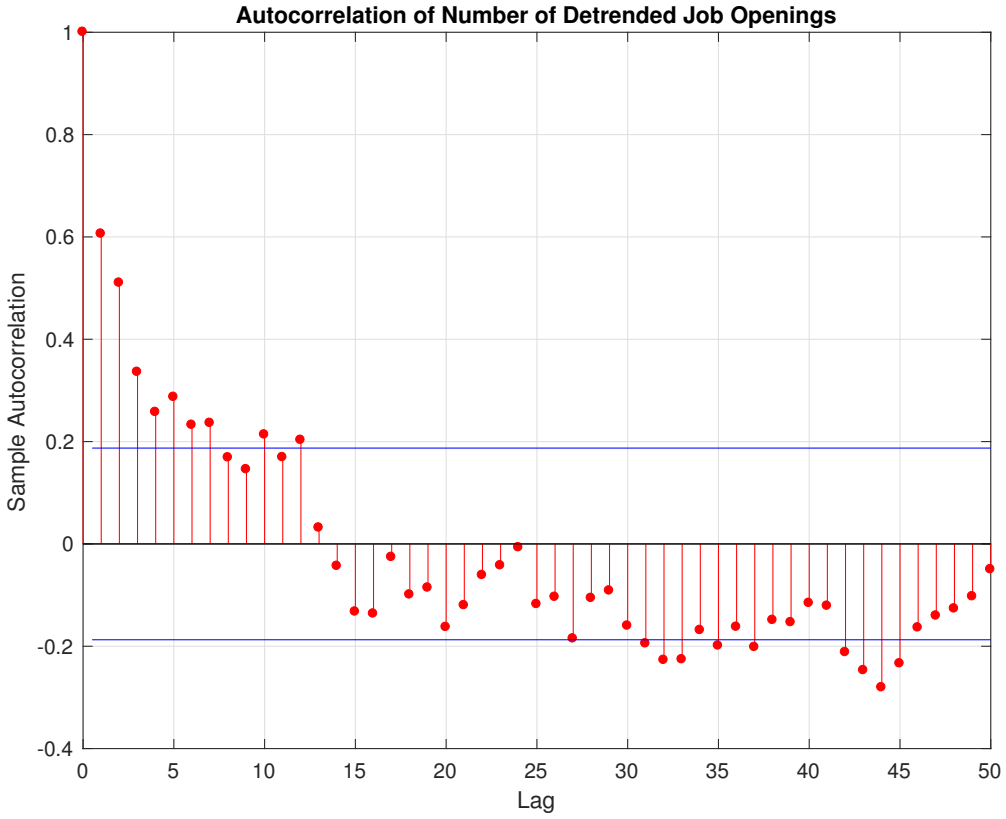
Figure 2: Monthly Data from 08/2007 to 01/2017.

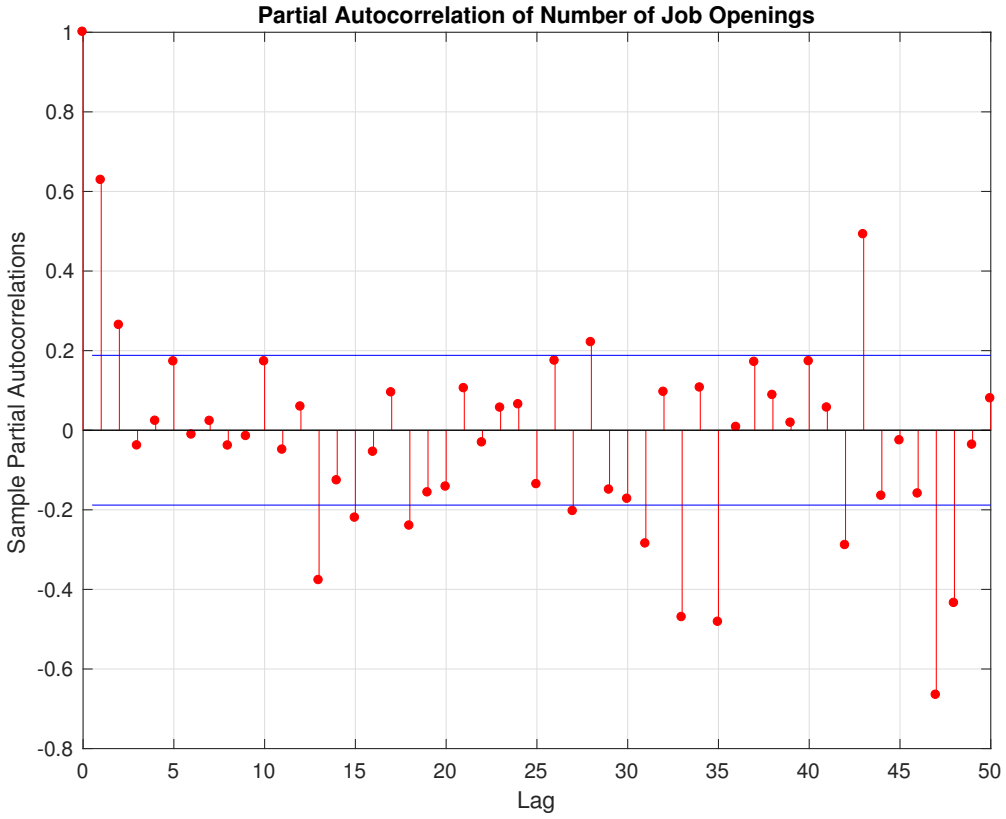


One can observe from Figure 2 that, in this senario. the detrended number of job openings is not high correlated with other major indexes, which in turn implies that in terms of modelling and forecasting, a univariate model may be better than a multivariate model. Such an implication can be first confirmed by the cross correlation plot given by Figure 2.2: the cross correlations are almost within the tolerable level.



Now let's focus on the detrended series $\hat{u}_t$ alone. Graphically it still exhibits a strong serial autocorrelation, which can be further consolidated by the autocorrelation Figure 2.2 and partial autocorrelation Figure 2.2 plots


Autocorrelation of Number of Detrended Job Openings
Sample Autocorrelation
Lag
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0
5
10
15
20
25
30
35
40
45
50



Compare to the autocorrelation and partial autocorrelation plots for non-detrended series Figure 2.1.1 and Figure 2.1.1, one can observe that the autocorrelation of detrended series quickly vanishes after only a few lags, up to certain tolerable level. Apparently, a certain type of time series model should be adopted to model the detrended series.

2.2.1 Univariate Modelling: ARMA

Now that the above subsubsection demonstrates that proper time series models should be adopted to model $\hat{u}_t$, a natural question is which time series models to use?

Two natural choices are *Autoregressive-moving average* (ARMA) processes and its extension *autoregressive-integrated moving average* (ARIMA) processes. Rationale from time series analysis tells us that if the data

- exhibits no apparent deviations from stationarity, and
- has a rapidly decreasing autocorrelation function,

then a suitable ARMA process would be good choice. However, if any of the two properties is violated, we shall first look for a transformation of the original data

which generates a new series with the above properties. Such a transformation is often achieved by difference, leading us to ARIMA models.

We have seen how the autocorrelation of series varies with different lags from both Figure 2.2 and Figure 2.2. Therefore the second property of ARMA model listed above is checked. Now we focus on checking whether the three series are stationary, which is a rather fundamental question when dealing with time series data. We apply the two most commonly used statistical tests for checking if a time series is a unit root process or stationary:

- Augmented Dickey-Fuller (ADF) test for unit root
- Phillips-Perron test for one unit root

The test results are presented in Table 1. Throughout the three series, both returns and squared returns reject the null hypothesis of unit root and thus imply stationary. Consequently, we can comfortably use ARMA-type models to approach volatility processes.

Table 1: Stationarity Test for $\hat{u}_t$

	Augmented Dickey-Fuller Test	Phillips-Perron Test
Detrended No. Job Openings	1	1

"1" represents the data is stationary while "0" is not.

Now we are legal to we are legal use the AMRA(m, n) models of the following form as data generating process: recall that u_t is the detrended number of job openings at month t, then we assume u_t to be an Autoregressive Moving Average process

$$u_t = u + \epsilon_t + \sum_{i=1}^{p} \psi_i u_{t-i} + \sum_{i=1}^{q} \theta_i \epsilon_{t-i} \tag{2.2.1}$$

where u is the long-run unconditional mean, p and q are the orders of autoregressive and moving average terms of ARMA process respectively. Here we just assume that ϵ_t follows Gaussian distribution:

$$\epsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2),$$

for some unknown variance σ^2.

Now we proceed to select the best model of ARMA(p, q). More specifically, what we would like to do is to pin down the orders p and q. In this project, we use the Schwartz Bayesian information criterion (SBIC) to implement the model selection and Akaike information criterion (AIC). We consider a range of p and q from 0 to 7, which would be broad enough to cover most cases.

```
BIC for p,q = 0,...,7
315.5228  319.5900  344.4571  352.1965  356.9533  364.3398  365.0868
289.6255  294.3894  305.5522  310.1598  312.7959  319.4566  322.3869
295.0268  321.8883  326.6763  331.0932  334.6805  339.8552  343.4834
324.9045  331.1503  351.0602  357.9553  361.2584  366.2077  369.6783
330.5317  331.3058  360.3732  367.7483  371.6577  377.9385  379.7419
331.7363  337.1871  361.6146  370.2161  375.1831  380.3524  384.3523
341.2668  344.5294  363.9704  378.0267  382.4437  387.9619  390.9684
343.2756  348.3916  372.6336  380.8600  386.5527  392.2557  397.4499

AIC for p,q = 0,...,7
310.2534  311.6859  333.9182  339.0228  341.1449  345.8967  344.0089
281.7213  283.8504  292.3786  294.3514  294.3528  298.3787  298.6744
284.4879  308.7146  310.8680  312.6501  313.6027  316.1426  317.1361
311.7308  315.3420  332.6171  336.8775  337.5458  339.8604  340.6963
314.7233  312.8627  339.2954  344.0357  345.3105  348.9565  348.1252
313.2932  316.1092  337.9020  343.8688  346.2011  348.7356  350.1008
320.1890  320.8169  337.6232  349.0446  350.8269  353.7105  354.0822
319.5630  322.0443  343.6515  349.2433  352.3012  355.3695  357.9289
```

One can easily locate that for both BIC and AIC, ARMA(1,1) model is the best, the resulting estimated ARMA(1,1) model is hence given by

```
ARIMA(1,0,1) Model:
--------------------
Conditional Probability Distribution: Gaussian

Standard          t
Parameter       Value          Error       Statistic
-----------   -----------   ------------   -----------
Constant      -0.012699      0.0507761      -0.250098
AR{1}         0.845123       0.07602         11.1171
MA{1}         -0.355306      0.116155       -3.05889
Variance      0.657527       0.0820154       8.01712
```

Hence we conclude that

$$u_t = -0.012699 + \epsilon_t + 0.845123 u_{t-i} + -0.355306 \epsilon_{t-i} \tag{2.2.2}$$

2.2.2 Multivariate Model

To further justify the conclusion that univariate ARMA(1,1) model is sufficient to model the detrended series $\hat{u}_t$. We select a best multivariate model, from a couple of

candidates of Vector Autoregressive models (VAR), where we incorporate unemployment rate, fed rate and CIP. Then we compare the performance of the fitted VAR model with that of the univariate ARMA(1,1) model 2.2.2 obtained before.

For the sake of illustration, in this project we consider four different models for the data, from which we would like to choose the best one:

- VAR(2) with diagonal autoregressive and covariance matrices
- VAR(2) with full autoregressive and covariance matrices
- VAR(4) with diagonal autoregressive and covariance matrices
- VAR(4) with full autoregressive and covariance matrices

We fit our data to each of the above four models. To assess the quality of the models, we divide the data (matrix) series into three periods: presample, estimation (training set), and forecast (testing set). We first fit the models to the estimation data, using the presample period to provide lagged data. Then we compare the predictions of the fitted models to the forecast data. Here we emphasize that the estimation period is in sample, and the forecast period is "out of sample", which is usually called backtesting.

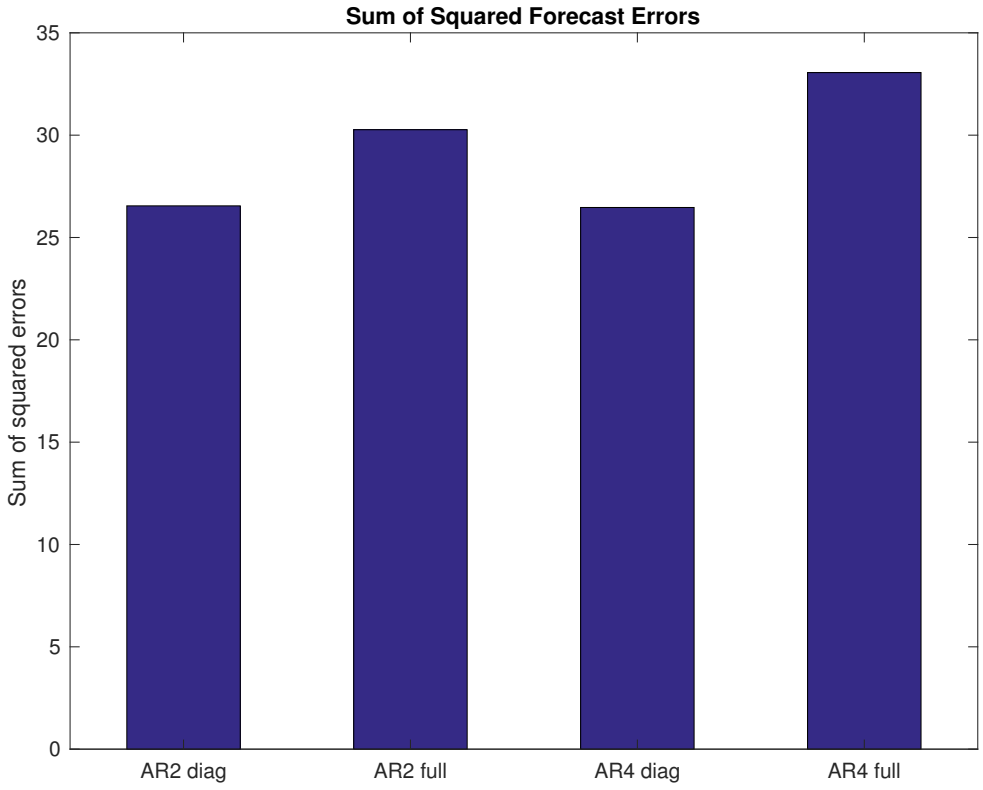
More specifically, for the two VAR(4) models, the presample period is the first four rows of the data matrix. We use the same presample period for the VAR(2) models so that all the four models are fit to the same data, which is necessary for model fit comparisons. For both models, the forecast period is the final 10% of the rows of original data matrix. So the estimation period for the models goes from row 5 to the 90% row.

We compare the predictions of the four models against the known testing data, where We calculate the sum-of-squares error between the predictions and testing data. The SSR across the four models are shown as in Figure 3

One can see from the Figure 3 that VAR(4) with diagonal autoregressive and covariance matrices is the best among the four. The estimation results is given as follows.

```
Model  : 4-D VAR(4) with Additive Constant
Conditional mean is AR-stable and is MA-invertible
Series : Job Count
Series : Unemployment Rate
Series : Fed Rate
Series : CPI
a Constant:
0.0298765
0.00928781
0.15921
0.171948
AR(1) Autoregression Matrix:
0.443469              0              0              0
```

Figure 3: Sum-of-squared Errors Across Four Models

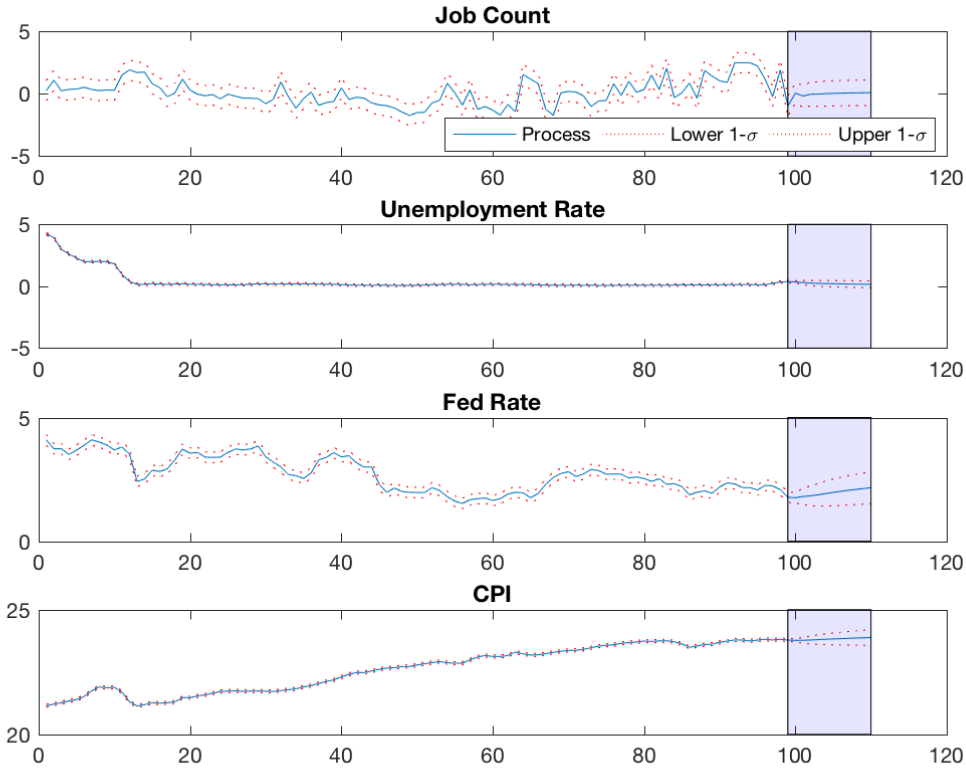



```
0        1.36825              0              0
0              0        1.18041              0
0              0              0        1.57918
AR(2) Autoregression Matrix:
0.239694              0              0              0
0      -0.548229              0              0
0              0      -0.327379              0
0              0              0      -0.741245
AR(3) Autoregression Matrix:
8.83881e-05              0              0              0
0     -0.0105543              0              0
0              0       0.179703              0
0              0              0       0.120817
AR(4) Autoregression Matrix:
-0.000678245              0              0              0
0      0.0997751              0              0
```

```
0               0     -0.0971763               0
0               0              0        0.034339
Q Innovations Covariance:
0.65493               0              0               0
0     0.00981125              0               0
0               0      0.0451541               0
0               0              0      0.00381126
```

We also visualize the out-of-sample forecasting based on the best VAR(4)-diagonal as in Figure 4

Figure 4: Backtest Prediction of AR(4) Model



2.2.3 Univariate v.s. Multivariate

Now that we obtain a best univariate 2.2.2 and the VAR(4)-diagonal as above, the use the out-of-sample testing again to compare the two models. It turns out that the sum-of-squares error for univariate ARMA(1,1) (2.2.2) is

$$23.7960,$$

while that for VAR(4)-diagonal is

$$26.4659.$$

Hence we further justify that the univariate model suffices to predict the stationary part of the number of job openings series.

2.2.4 Robustness Checking

Here we note that the ARMA model selection result, namely, ARMA(1,1), is robust to t-distribution for the errors ϵ_t.

```
BIC for p,q = 0,...,7
315.5228  319.5900  344.4571  352.1965  356.9533  364.3398  365.0868
286.3403  292.8852  301.7340  306.3183  309.3241  315.5814  319.4430
293.8424  321.2948  326.1316  330.4221  334.0922  339.1865  342.9961
324.9120  330.7510  351.0464  357.9763  361.2880  366.2137  369.6901
330.4557  330.6538  360.4055  367.7336  371.6923  377.9511  379.7350
331.7194  337.1724  361.6654  370.2551  375.1998  380.3821  384.3656
341.2755  344.4323  363.7749  378.0320  382.4585  387.8240  390.9760
343.2237  348.2244  372.6194  380.7385  386.5420  392.2224  397.4364


AIC for p,q = 0,...,7
357.9155  357.9155  357.9155  357.9155  357.9155  357.9155  357.9155
278.4361  282.3463  288.5603  290.5099  290.8810  294.5035  295.7304
283.3034  308.1212  310.3233  311.9790  313.0144  315.4740  316.6488
311.7384  314.9427  332.6033  336.8985  337.5754  339.8664  340.7080
314.6473  312.2107  339.3277  344.0211  345.3450  348.9691  348.1183
313.2763  316.0946  337.9528  343.9078  346.2178  348.7653  350.1141
320.1976  320.7197  337.4276  349.0500  350.8418  353.5726  354.0898
319.5112  321.8771  343.6373  349.1218  352.2905  355.3362  357.9155
```

One can still see that ARMA(1,1) is the best.

3 Category-Specific Job Growth

Though overall per-month job growth may be best predicted by the previous month's job growth, it is reasonable to suspect that job growth *per industry* or *per job type* may depend on more nuanced factors. To explore this speculation, we first used performed unsupervised clustering on the descriptions of job postings to identify novel job types. We then defined a univariate score of job growth that can be calculated per-city. Then, treating cities as observational units, we constructed regression models for each of our discovered job categories with job growth score as a response, and city demographic data as covariates. Our analysis revealed the strongest predictors of job growth within job types.

3.1 Unsupervised Learning of Job Types

We use the job description text data to cluster job categories. Each job categories has typically thousands of job postings which each have a text description. We use the spherical K-means algorithm to cluster the term frequency, inverse document frequency (TF-IDF) matrix where documents are the concatenation of all job descriptions within a job category. We describe our procedure in detail below and include supporting statistical justification for choices we made.

3.1.1 Data representation

Often the key part of natural language processing is data object representation; how to turn text data into numerical data. Once we have a rectangular data matrix (rows are observations, columns are variables) we can use standard statistical procedures to do clustering.

Our analysis is based on the document-term matrix (DTM) where the rows correspond to job categories and the columns correspond to all words that show up in our text data (there were 11,777 unique words).

In this case documents correspond to job categories. In particular, the document corresponding to each category is formed by aggregating all job descriptions within that category. The entries of the document term matrix are then the word counts in each category i.e. the i, j-th entry of the DTM is the number of times the j-th word shows up in the ith category (this is the so called bag of words representation because it ignores word order).

There are N=148 job categories and $d = 11,777$ unique words in the job description data. The document term matrix is then $148 \times 11,777$. This means each job category is represented by a vector of word counts in $R^1 1,777$.

It is well known in the natural language processing community that the raw word counts can be problematic as features. For example, words like the and to show up very frequently, but are not meaningful. One way to get around this by manually removing a hand curated list of commonly occurring words (so called stop word). This procedure, however, requires a lot of time and can often ignore features of a particular data set.

A more effective automatic feature engineering procedure is to downweight words that appear in many of the documents. The document frequency of a word is the number of documents that word shows up in (e.g. in our context the document frequency is an integer between 1 and 148). The inverse document frequency of a word W is then

$$\text{idf}(\mathbf{W}) = \ln\left(\frac{\text{total number of documents}}{\text{number of documents containing } \mathbf{W}}\right)$$

Notice that the larger the document frequency, the smaller the inverse document frequency. Now for a given document (D), word (W) pairing define the term frequency, inverse document frequency as

$$\text{tf-idf}_D(\mathbf{W}) = \text{tf}_D(W)idf(W)$$

where $\text{tf}_D(W)$ is the term frequency of W i.e. the number of times word W shows up in document D.

We now use the TF-IDF matrix as our data object representation i.e. the i-th document is represented by the d dimensional vector of its tf-idf scores.

One additional data representation choice we made was to first stem each word (using the porter stemmer implemented in the snowballIC library in R). Stemming attempts to reduce each word to its base form. Without stemming words like consultant and consultants are considered as two distinct words. Stemming consultants will remove the s at the end and map consultants to consultant. Before stemming there were 11,777 unique words and after stemming there were 7,727 unique words.

3.2 Clustering algorithm choice

There are two important characteristics of the TF-IDF matrix to note which inform our choice of clustering algorithm. First, the TF-IDF matrix is sparse (most words only show up in a few documents). Sparsity allows us to use custom data structures and linear algebra algorithms which significantly reduce the time and computer memory resources required. The second characteristic to note is that research in natural language processing has shown the cosine distance is better than the raw euclidean distance as a measure of similarity between two tf-idf vectors. The cosine distance between two vectors is the angle between them. This distance measure corresponds to projecting the data onto a sphere in $R^1 1,777$.

Based on the two observations in the above paragraph we choose decided to use the spherical K-means algorithm (SKM) to cluster documents tf-idf vectors. SKM is essentially K-means operating on a sphere; when computing the distance between a point and a cluster centroid spherical distance is used (e.g. cosine similarity) as opposed to euclidean distance (this is an example of manifold learning). Additionally, we choose the skmeans package in R because it can operate on sparse matrices.

3.2.1 Clustering results

We choose $K = 12$ for simplicity and interpretability. The 12 job type clusters are listed in full in Appendix A. We manually specified names of these 12 job types based on the constitutent job categories. In Figure 5, we display a sample "word cloud" from job descriptions of job categories in the "Business" cluster:

As a preliminary analysis of job growth by job type, we plotted the job posting counts per month, separated by the discovered job types. This plot is shown in Figure 6.

Figure 5: Word cloud of job descriptions from the "Business" cluster.

3.3 Job-Growth Score

Importantly, our analysis in this section does not employ time-series models. Thus, to perform regression, we needed to encapsulate the job growth (in a particular city, of a particular discovered job type) with a univariate metric. In this section, we discuss two different scores we attempted in our analysis, and their strengths and weaknesses.

3.3.1 Regression slope score

An intuitive way to measure job growth over a certain span of time is to calculate the linear regression coefficient of job counts per month (as a response) vs. the month index. Even if the job growth trend is non-linear, positive association will still be reflected in the estimated slope. This is shown in Figure 7, which shows new job postings as a function month index.

3.3.2 Dot product score

We found that the regression slope score did not adequately measure job growth for all types of job growth curves. For instance, if a city experienced *no* job growth for the first half of the time range, large growth for the third quarter, and small but non-zero growth for the final quarter, the regression slope would be (roughly) positive. However, that city would nonetheless be experiencing job posting *decrease* near the end of the time range.

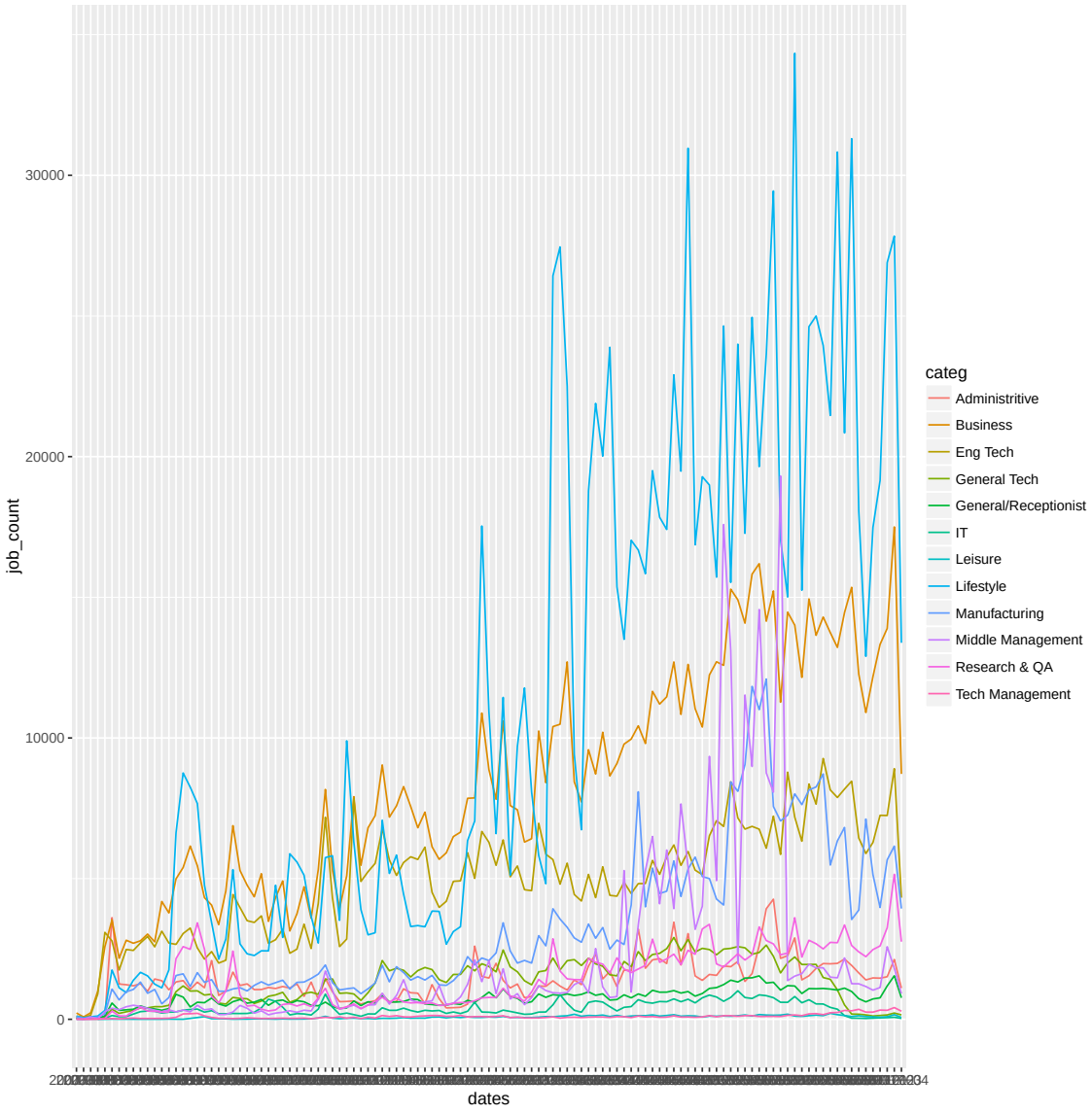


Figure 6: Job growth by discovered job type.

Thus, we decided to weight month-wise job posting counts by month recency. We term this the "dot-product" score. Specifically, let i index months, and j_i be the job count for month i then the dot-product score is written as follows:

$$S = \sum_{i=1}^{\#\text{ months}} i \cdot j_i$$

In general, we found that use of this score resulted in more sensible results, which we describe in the next section.

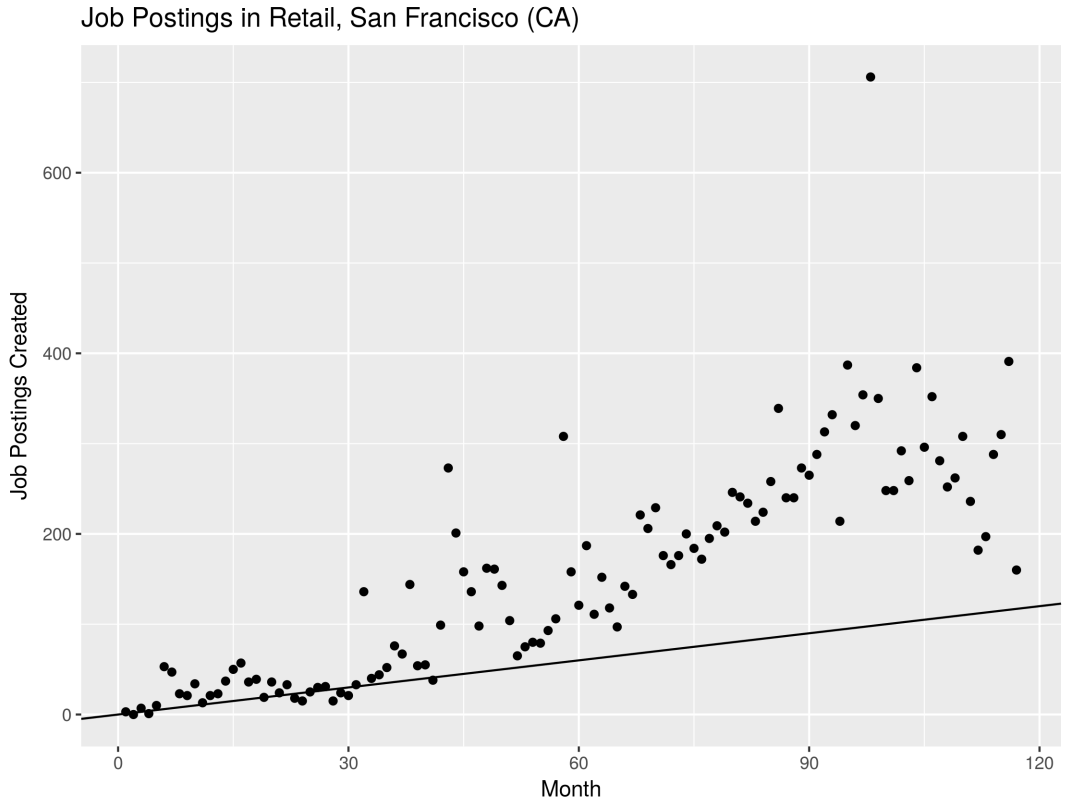


Figure 7: Retail job listing counts per month in San Francisco, CA

3.4 Regression Analysis and Results

In this section we present results from job-type-wise regression of city job growth scores on city covariates. Two sets of 12 regressions were performed; the first set involved the Regression Slope Score as the response, and the second set used the Dot Product Score as the response. The 12 regressions with each score correspond to the job types discovered by the analysis presented in Section 3.1.

Our results are presented in Figures **??** and 9. We found more interpretable results using the dot product score. For instance, high dot product growth scores for technical fields depended strongly on the cities having high numbers of graduate and college level degrees within their populations.

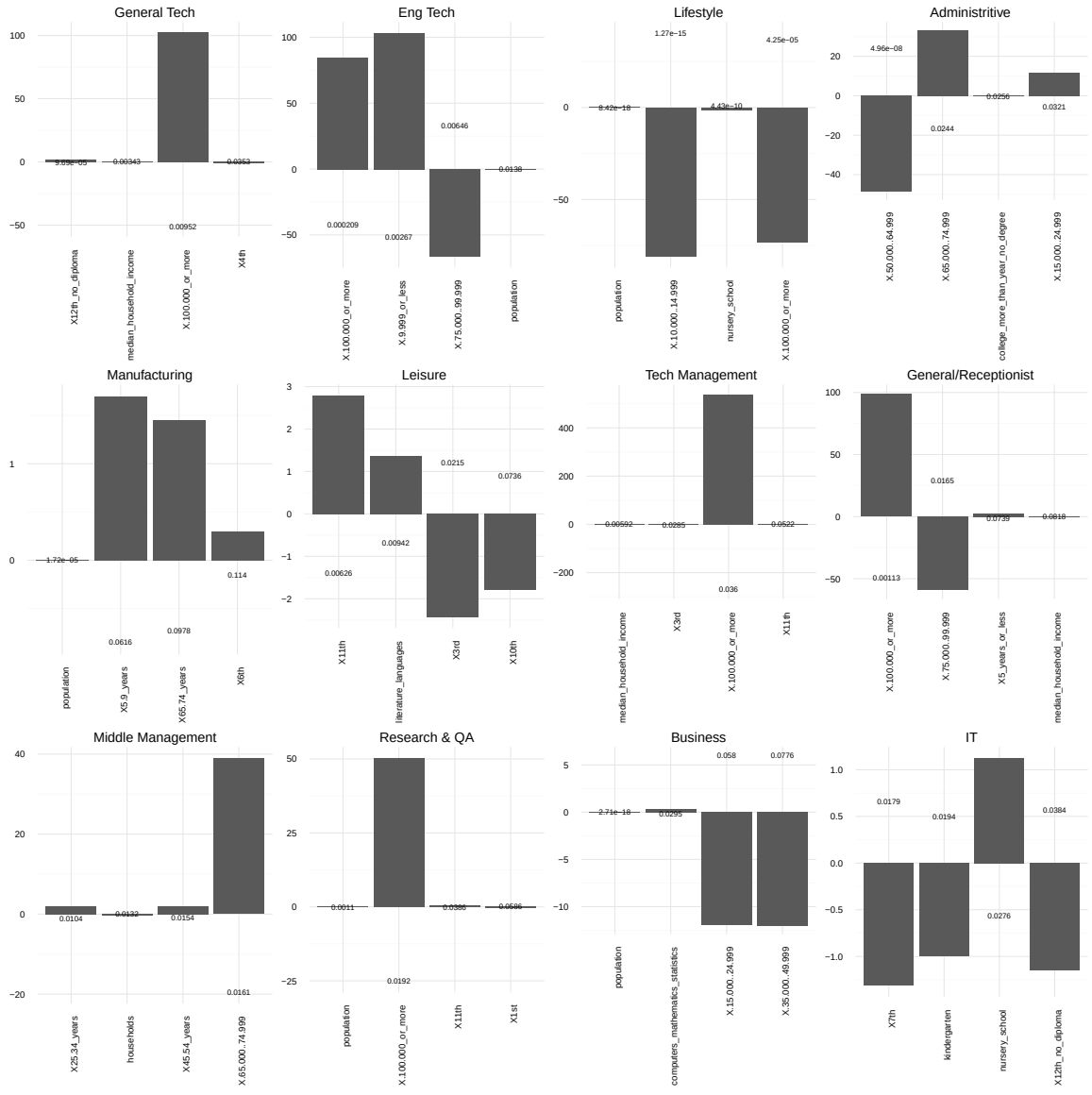


Figure 8: labelfig:reg1 Regression results with regression slope score.

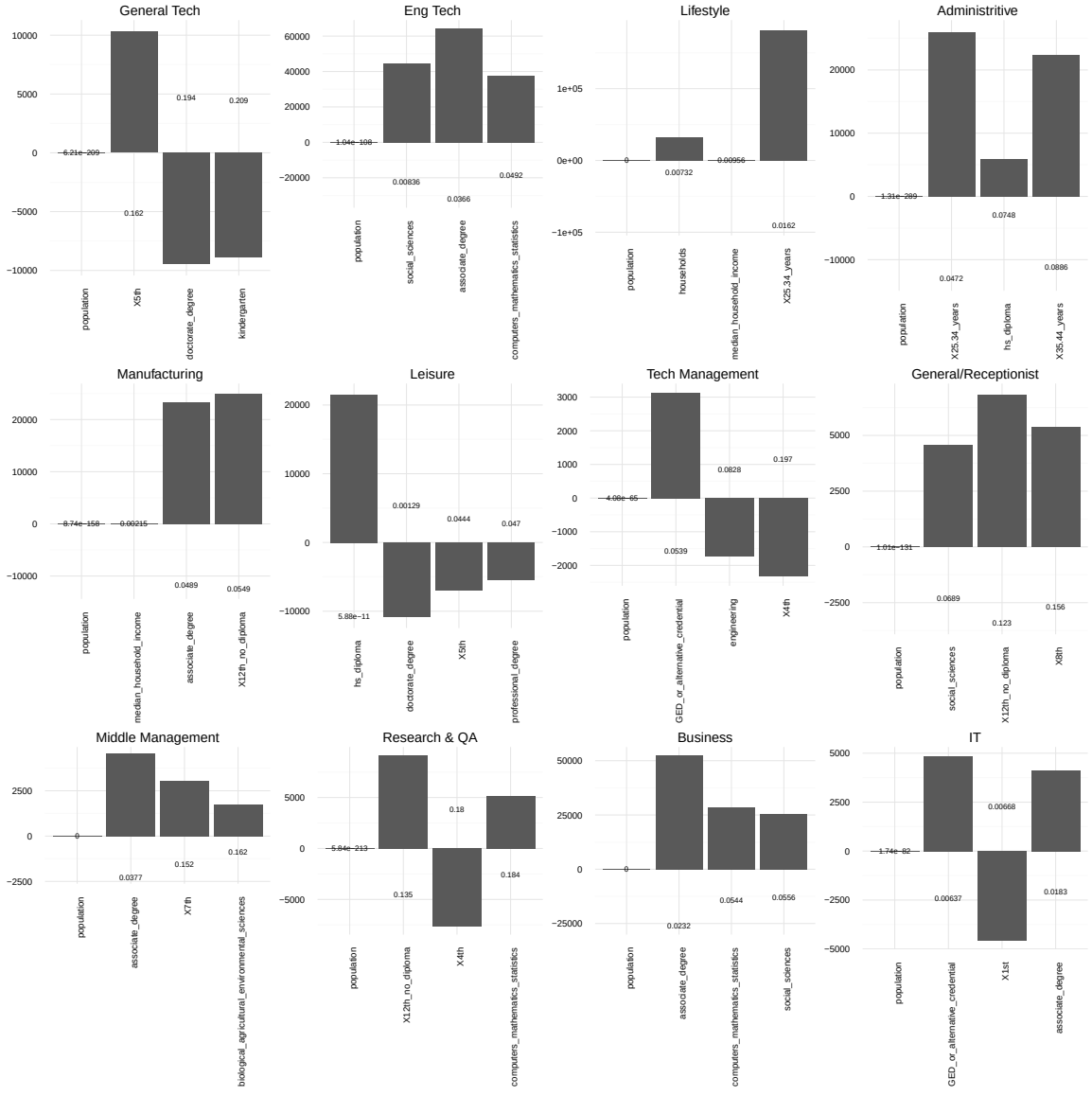


Figure 9: Regression results with dot product score.

3.5 Future work

Next steps in this analysis would involve careful validation and a rigorous model selection process.

4 Conclusion

Our analysis provided two diverse perspectives on job growth in the U.S. since the Great Recession. First, we provided a careful time-series analysis showing that per-month job growth is best predicted by job growth in the previous month. Then, we discovered interpretable job-types using document clustering, and showed that job growth within-type depends on intuitive features of the population.

A Appendix: Job Types from Spherical Clustering

‘General technology‘
1 ”Technology”
”Tech Management”
”Web Development”
4 ”Recruiting”
”Government”
”Software Architecture”
7 ”Executive Management”
”Lab Technician”

‘Engineering technology‘
1 ”Technical Design”
”Software Development”
3 ”Biological Sciences”
”Computers & Hardware”
5 ”Social Services & Mental Health”
”Software, Gaming & Web Developers”
7 ”Nursing”
”TV, Film & Video”

Lifestyle
1 ”Life, Physical, and Social Science”
”Retail”
3 ”General Management & Business”
”Sales & Business Development”
5 ”Training & Instructor”
”Shipping/Receiving”

7 "Agriculture, Forestry & Fishing"
"Marketing"
9 "Intern / New Graduate"
"Arts, Media & Publishing"
11 "Salon/Spa/Fitness"

Administrative
1 "Office Manager"
"Financial Services"
3 "Job Fairs"
"Customer Service"
5 "Banquet, Catering & Events"
"Inventory"

Manufacturing
1 "Manufacturing & Operations"
"Tech Quality Assurance"
3 "Warehousing"
"Product Marketing"
5 "Restaurant & Food Service"
"Writing & Editing"
7 "Engineering & Architecture"

Leisure
1 "Travel & Tourism"
"Concierge & Guest Services"

'Technology Management' 1 "Engineers"
"Management Consulting"
"IT Operations"

Clerical
1 "Science, Pharmaceutical & Biotech"
"Administrative Assistant"
3 "Bookkeeping"
"Admin & Clerical"
5 "Receptionist"
"Telecommunications"

'Middle Management'
1 "Insurance"
"Transportation"
"Finance Management" 4 "Legal"

"Education & Training"

'Research and Q/A'
1 "Maintenance & Repair"
"Operations"
"Research"
4 "Logistics"
"Plant Management"
"Trading"
7 "Other Healthcare"
Business
1 "Accounting & Finance"
"Business Development"
3 "Health & Medical"
"Construction & Skilled Trade"
5 "Sales Rep"
"Supply Chain & Logistics"
7 "Healthcare Management & Finance"
"Human Resources"
9 "Law Enforcement & Security"
"Account Management"
11 "Purchasing"
"Computer Systems Support"
13 "HR Management"
"Mathematical"
15 "Executive Assistant"
"Information & Data Analytics"

IT
1 "System Admininistrator"
"Network Administrator"
"Direct Sales"
4 "Technical Support"
"Public Relations"