The Structure of Flight Delay Networks

Landon Buechner LRBUECHNER17@TAMU.EDU
Department of Mathematics
Texas AM University
College Station, TX 77840, USA

Editor: Suhasini Subba Rao

Abstract

The pairwise relationships between airports and their delays can be represented as a graphical model with airports as nodes. In this work I develop the theory behind gaussian graphical models, discuss how the fused lasso can segment a time series, and fit a piece wise stationary 10 dimensional VAR(1) model to the third quarter of 2018.

1. Introduction

In the airline industry, it is important to understand the behavior of flight departure delays. This knowledge enables both customer and airline company to make more informed decisions about flight planning. One might claim that intuitively, nearby airports have similar average delays due to localized factors such as weather and other exogenous variables. While this may be true, it is of interest to consider the relationship of between airports themselves. For example, if the delays out of IAH are high at any given time, what can be said about LAX? More specifically, during a given season, what airports are more correlated to one another?

This relationship between airports can be abstracted and represented in graphical form with each node as a unique airport. First, key concepts about multivariate Gaussian distribution, Markov random fields, and properties of the precision matrix are introduced. With the probabilistic foundations and notation established, I introduced the fused graphical lasso and it's potential uses for time series segmentation.

After motivating the models used, I begin my analysis with a description of the data cleaning procedure followed by a case study of American Airlines. I conclude the analysis by explaining how I used the fused lasso in conjunction with a 30 dimension VAR(1) model to predict average daily departure delays for the 3rd quarter of 2019.

2. Modelling

2.1 Gaussian Graphical Models Graphs

Knowing that the transformed average daily delays are normally distributed implies that they can be expressed as a centered multivariate gaussian distribution. Let the random vector $\underline{Y}' = (X_1, \dots, X_d)$ denote the log average daily delay times for airports $\{1, \dots, d\}$ be jointly normal with covariance matrix Σ and inverse covariance matrix Ω (Usually refered to as the precision matrix). For now, assume that at a population level, the relationships between airports doesn't change as a function of time. This has the joint density

$$f(X) = (2\pi)^{-d/2}(\text{det}\Sigma)^{-\frac{1}{2}}\text{exp}\bigg\{-\frac{1}{2}\underline{Y}'\Omega\underline{Y}\bigg\} \tag{1}$$

Define $\mathcal{G} = (\mathcal{A}, \mathcal{E})$ as an undirected graph where $\mathcal{A} = \{1, \dots, d\}$ is the set of airport nodes and $\mathcal{E}$ is the set of edges such that for a pair of airports i and j, $(i, j) \in \mathcal{E}$. $\mathcal{G}$ is classified as a Markov random field if the nodes satisfy the Markov properties. The main property of interest is pairwise Markovity. The density $f(*)$ satisfies pairwise Markovity with respect to $\mathcal{G}$ if for a pair of random variables X_i, X_j and remaining airports $X_{\mathcal{A}\backslash\{i,j\}}$

$$f(X_i, X_j|X_{\mathcal{A}\backslash\{i,j\}}) = f(X_i|X_{\mathcal{A}\backslash\{i,j\}})f(X_j|X_{\mathcal{A}\backslash\{i,j\}}) \tag{2}$$

Thus a pair of airports are pairwise Markov if any only if they are conditionally conditionally independent. Thought about in the context of a Guassian graph, conditional independence implies no edge between i and j. At a population level, this is encoded as a 0 in the precision matrix, namely $\Omega_{ij} = 0$. Conversely, two airports are dependent if they are not pairwise Markov which implies that they are connected by an edge in $\mathcal{G}$. Contrast this with the fact that Σ_{ij} for a multivariate Gaussian implies pairwise independence.

In order to estimate the underlying graph structure of $\mathcal{G}$, and in particular identify when $\Omega_{ij} = 0$ we state the following well known relationship. Let $\underline{Y}_{\mathcal{A}\backslash\{i\}} \in \mathbb{R}^{d-1}$ such that the entry X_i is omitted. If $\underline{Y}_{\mathcal{A}\setminus\{i\}}$ is regressed onto X_i where ϵ_i is independent of $\underline{Y}_{\mathcal{A}\setminus\{i\}}$, then $\beta_{i,j}$ if and only if $\Omega_{i,j}$. Thus, the zero valued parameter estimates correspond to zeroes in the precision matrix.

$$X_i = \sum_{j \notin \mathcal{A}} X_j \beta_{i,j} + \epsilon_i \tag{3}$$

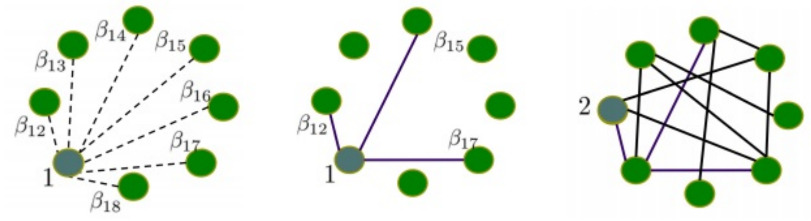
Define $\underline{Y}_t' = (X_{t,1}, \ldots, X_{t,d})'$ as the vector of log delay times at time t. Assume that the marginal of $\{\underline{Y}_t\}_{t=1}^T$ is a centered multivariate Gaussian with variance Σ. This implies that the distribution of $\underline{Y}_t$ is independent from time t and has constant mean and variance for all t. In reality, this model has unrealistic assumptions that do not take into account seasonal relationships between airports. The fused lasso discussed in the following section is a natural extension of this model that allows for time varying parameter estimates.

For each node, the estimation procedure is as follows. An ℓ_1 penalty is applied to encourage sparsity in the parameter estimates.

$$\hat{\boldsymbol{\beta}}_i = \arg\min_{\beta} \left\{ \frac{1}{2T} \sum_{t=1}^{T} \left(\underline{Y}_t - \boldsymbol{\beta}' \underline{Y}_{\mathcal{A}\setminus\{i\}} \right) + \lambda \left\| \boldsymbol{\beta} \right\|_1 \right\} \tag{4}$$

After estimating the set of parameters $\{\hat{\boldsymbol{\beta}}_i\}_{i=1}^d$, stitch the graph together by taking the maximum of the pairwise estimates to obtain Ω. In the context of the problem at hand, this generates a network that encodes the strongest dependencies between airports.

$$(i,j) \in \hat{\mathcal{E}} \Leftrightarrow \max\left\{ \hat{\beta}_{i,j}, \hat{\beta}_{j,i} \right\} > 0.01 \tag{5}$$



(a) LASSO regression for node 1 (b) Edge selection of node 1 (c) LASSO for all nodes

Figure 1: Neighborhood estimation

2.2 Fused Lasso

Now that foundations for Gaussian graphical models have been established, consider a graphical model that changes over time. The model discussed above assumes the contemporaneous variance of Y_t does not evolve over the observed time interval $t = 1, \ldots, T$. In the context of airline delay networks, this is an obvious over simplification because seasonal travel behaviors evolve over the course of a year. For example, delays during the summer months are a function of summer specific events such as increased travel to vacation destinations and thunderstorms. In order to estimate how the network changes over time, the graphical model discussed above need to be slightly modified.

The Fused Lasso estimates a precision matrix for each time step t subject the constraint that they are piece wise constant and are sparse. In other words, instead of estimating one precision matrix Ω as discussed above, the fused lasso estimates a sequence of precision matrices $\{\Omega_t\}_{t=1}^T$ and minimizes their sequential differences. Thought about in terms of one node, the parameter estimate for node i, $\boldsymbol{\beta}_i(t)$, should not be much different than $\boldsymbol{\beta}_i(t-1)$. Only at times, called breakpoints, when there is a significant change in the dependencies between airports should parameters change significantly.

Note that typically the fused lasso is used as a predictive model. My goal is not to make predictions, but to segment the time series into by identifying dates that signify structural changes in the graphical model. Since the fused lasso estimates a set of precision matrices that are piece wise constant, we can analyze their sequential differences and see for which dates the parameters change the most. In the limit; when λ_2 is very small, the precision matrix entries $\Omega_i j$ vary wildly from day to day and when λ_2 is very large, the estimates are forced to be constant for all values of t. Hypothetically, there should exist a λ_2 such that $\{\Omega_t\}_{t=1}^T$ are constant on unique time intervals and accurately estimate the true graph structure during that period. Within this framework, breakpoints can be identified and ultimately used to segment a multivariate time series.

The estimation procedure is done node by node. For a given airport i, a set of parameters $\{\hat{\boldsymbol{\beta}}_i(t)\}_{t=1}^T$ is estimated where $\hat{\boldsymbol{\beta}}_i(t) \in \mathbb{R}^{d-1}$ is a coefficient vector of the $d-1$ airports that aren't i. After estimating the T parameter estimates for each node, stitch together the precision matrices Ω_t for each date as prescribed in the previous section.

$$\underset{\{\hat{\boldsymbol{\beta}}_i(t)\}_{t=1}^{T}}{\arg\min} \left\{ \sum_{t=1}^{T} \left(\left\| \underline{Y}_{t,i} - \underline{Y}_{t,\mathcal{A}\setminus\{i\}} \boldsymbol{\beta}_i(t) \right\|_2^2 + \lambda_1 \left\| \boldsymbol{\beta}_i(t) \right\|_1 \right) + \lambda_2 \sum_{t=2}^{T} \left\| \boldsymbol{\beta}_i(t) - \boldsymbol{\beta}_i(t-1) \right\|_1 \right\} \tag{6}$$

The multipliers λ_1 and λ_2 are typically selected through a cross validation scheme informed by out of sample predictions. I chose an arbitrary λ_1 and made an informed choice for λ_2 that ensured the identified breakpoints weren't days apart but also not a whole year apart. It is reasonable to assume that structural relationships in the network persist for at least a few weeks and at most several months.

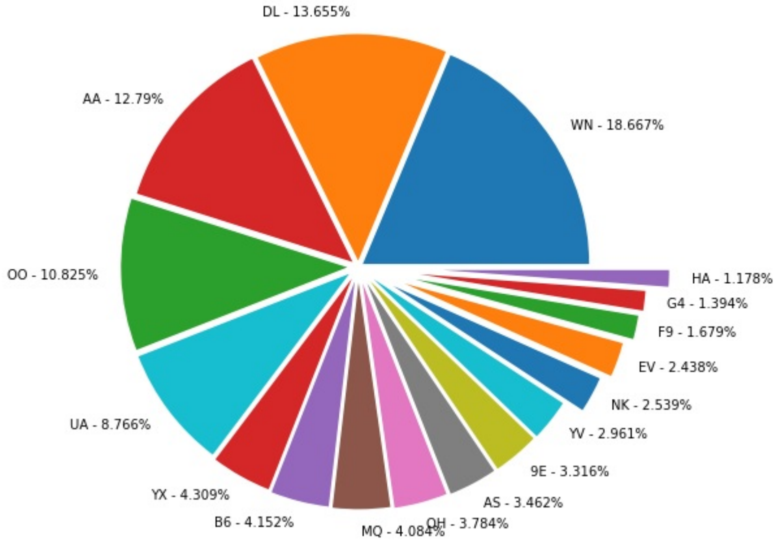
Unlike the traditional lasso regression, the fused lasso does is not available in python statistical libraries. I ended up using a well known convex optimization solver, cvxpy, to solve the objective function stated above. I conducted a simulation study in order to verify my code (See appendix).

3. Predicting Departure Delays

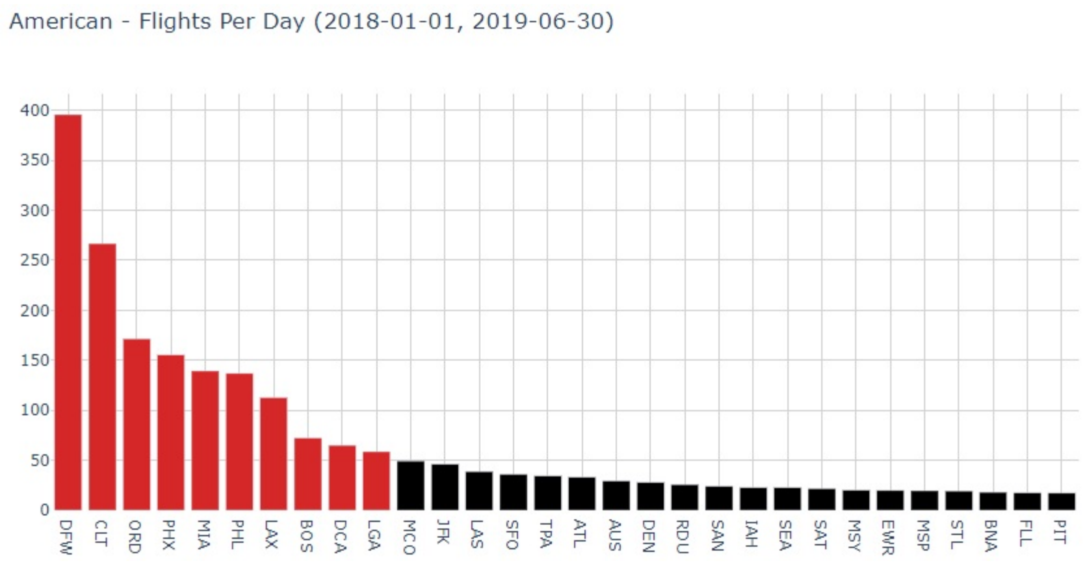
In following analysis, I conduct a case study of American Airlines and make predictions for the third quarter of 2019. I assume there is seasonality in the dependencies between airport delays. That is, the joint distribution of delay times in one year is the invariant of yearly shifts. Essentially, I assume that the structure of the delay network over some time interval, typically a couple of months, is approximately the same the next year. For example, it is reasonable to believe that Denver International Airport (DEN) and Colorado Springs Airports (COS) experience similar delays times in the winter due to the influx of skiers and snow storms.

Building upon this fact, I use fused lasso to identify dates in 2018 that signify a structural changes in the delay network. Based on these estimated breakpoints, I fit a piece wise stationary 10 dimensional VAR(1) model to the third quarter of 2018 and make predictions out of sample for the same period in 2019.

The given data set contains 17 unique airline carriers. Each of these companies offers different travel services to various locations in the united states. Thought about in the context of a network, each carrier has a unique geographic network that determines which airports are dependent on one another. For example, Conversely, Honolulu International Airport (HNL) and O'Haire International Airport (ORD) are likely to be uncorrelated due to localized factors. These are obvious examples and could be inferred without complex statistical models; what is interesting is pairs of airports that are highly dependent but not immediately obvious.



I choose American Airlines because they are the 3rd most active airline in terms of flight frequency and have flights across all of North America. Additionally, I sort the airports based on daily flight counts and retain the top 10 to use in the model. It makes sense to remove small airports from the model because they are almost surely independent of all other airports. In line with this idea, it turns out that in doing so most of the outlying delay times are removed.



3.1 Data Cleansing and Preliminary Analysis

Python's core data science ecosystem was used to conduct this study. Some noteworthy packages are

numpy	pandas	matplotlib
scipy	sklearn	statsmodels
plotly	cvxpy	networkx

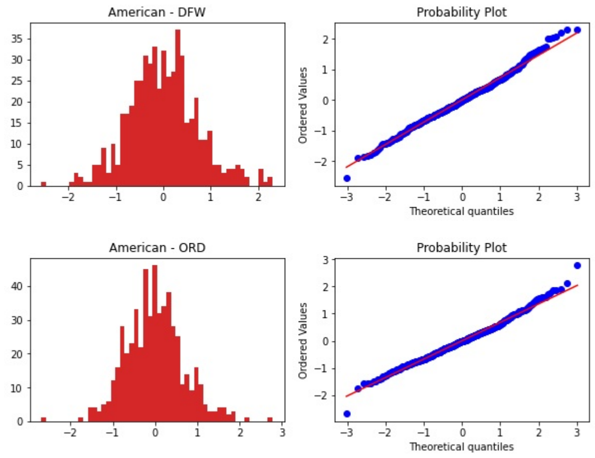
Below is a step by step description of the most important data cleaning steps. Reference the class called 'prep_delay_data' in order to gain a more nuanced understanding.

1. Airport information is geographic in nature so it important to be able to make meaningful plots of North America. An external data set is used to import latitude and longitude of all airports in the data set.

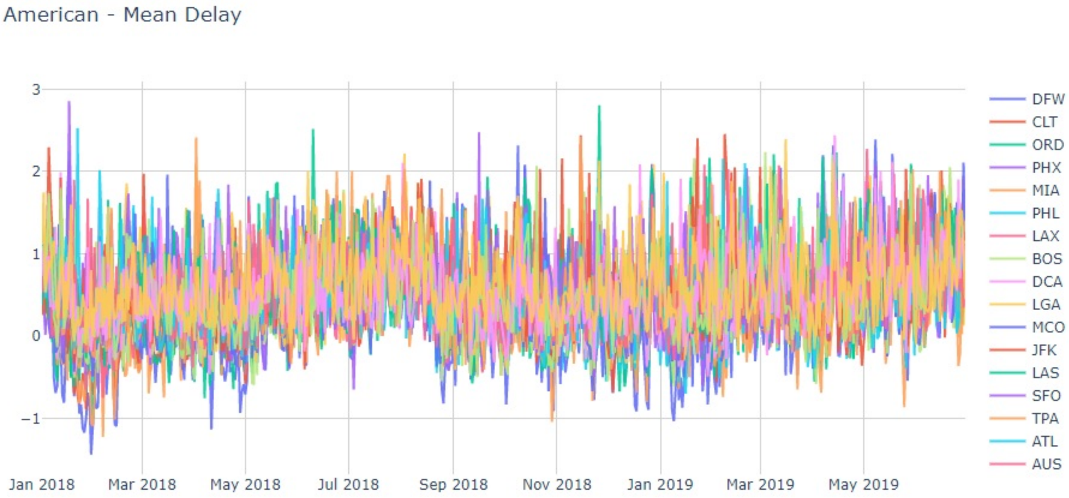
2. Discard flights with N/A departure delay information and cancelled flights.

3. For the top 10 airports, if on one day there are 0 flights out of a given airport, delete the average delay for all other airports on that day. This is to make sure the time series are of equal length. It turns out that for the main airline carriers this is not an issue.

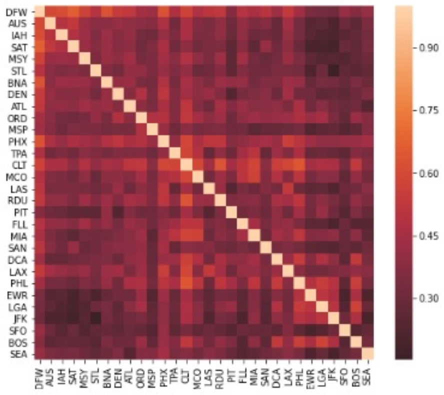
4. For airport $i = 1, \ldots, 10$, transform the time series of average daily delays individually. First calculate the minimum min_i and add $|min_i| + 1$ to the data. With this positive series, take the logarithm, calculate the mean $\mu_{i_{log}}$ and center the data. These transformation parameters are recorded in a table and used to convert back to minutes after making predictions with the transformed data. Centered data is essential because there is no intercept term in the fused lasso. It can be seen that each transformed series is in fact approximately normal.

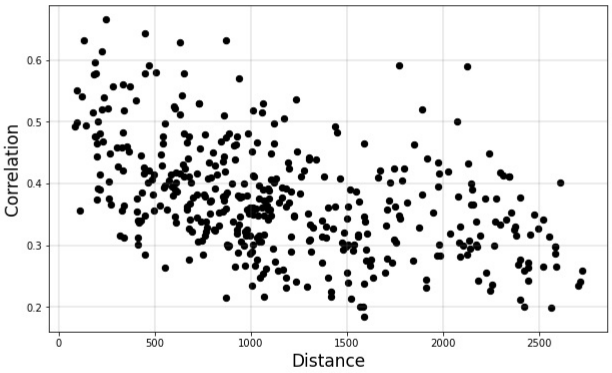


As mentioned before, a initial assumption is that delay dependence should be a function of distance. The correlation matrix supports this claim. For this calculation, I used 30



airports instead of 10 just to get a more complete picture of this interaction. The axes are ordered by first choosing the top airport for American Airlines, DFW, and sorting the rest based on their relative distance from DFW. Unsurprisingly, it turns out that there is a rough inverse linear relationship between distance and correlation. The horizontal axis in Figure 5, unlike the those of Figure 4, show pairwise distances between airports. This linear relationship is more pronounced for other airline companies.





Interestingly, it is seen that DFW and CLT have the highest correlation to the remaining airports. This hints that the average daily delay of all other airports might be dependent on the state of DFW and CLT. This is proved later by testing Granger causality. As stated in Section 2.1, we can naively assume that the dependencies between airports do not vary with time and estimate the precision matrix for the joint distribution of $\underline{Y}' = (X_1, \ldots, X_d)$. Recalling that the estimated precision matrix Ω encodes the graph structure, we uncover the undirected Gaussian graph.

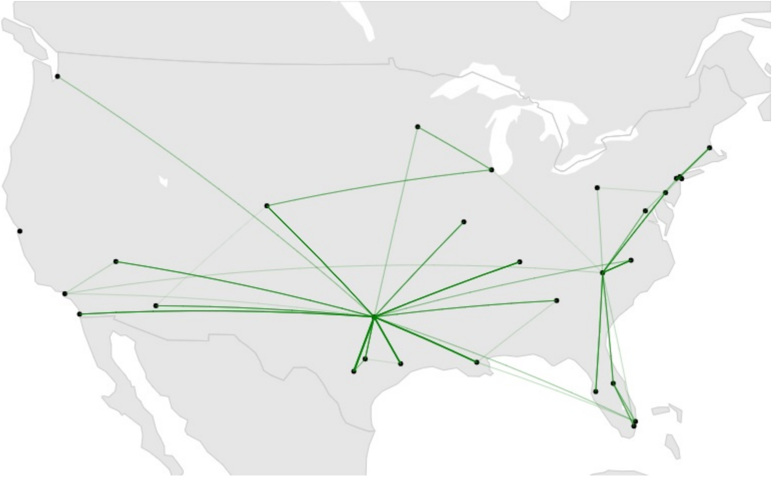


Figure 2: Dallas/Fort Worth (DFW) & Charlotte Douglas (CLT)

Incredibly, the Gaussian graphical model captures the conditional dependencies and shows that DFW and CLT are in fact related to most other airports. It can also be seen that airports that are nearby tend to be connected; supporting our intuition about distance and correlation.

3.2 Multivariate Time Series Segmentation

That the fused lasso allows the contemporaneous variance of Y_t to evolve over the observed time interval $t = 1, \dots, T$. A much needed condition in order to capture the seasonality of the the graph sturcture. This translates to estimating the state of the network, similar to the one above, at each time step. The power of the fused lasso is that it estimates these precision matrices (encoded networks) subject to the constraint that they are piece wise constant. This 'fuses' the precision matrices together such that the state of the network at time t is not much different than at $t-1$. Mathematically, let $\Omega_{\Delta t} = \Omega_t - \Omega_{t-1}$ be such that $\Omega_{\Delta t} \approx \Omega_0$ where Ω_0 is a 0 valued matrix.

$$\Omega_{\Delta t} = \begin{pmatrix} \Omega_{1,1} & \dots & \Omega_{1,d} \\ \vdots & \ddots & \vdots \\ \Omega_{d,1} & \dots & \Omega_{d,d} \end{pmatrix}_t - \begin{pmatrix} \Omega_{1,1} & \dots & \Omega_{1,d} \\ \vdots & \ddots & \vdots \\ \Omega_{d,1} & \dots & \Omega_{d,d} \end{pmatrix}_{t-1} \approx \begin{pmatrix} 0 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & 0 \end{pmatrix} \approx \Omega_0 \tag{7}$$

For a given time segment starting at t_0 of arbitrary length k, then $\forall t \in \{t_0, \dots, t_k\}$ $\underline{\Omega_i}_{\Delta t} \approx \Omega_0$. Stated equivalently in terms of a specific airport i with edges encoded in the row vector $\underline{\Omega_i}_t = (\Omega_{i,1}, \dots, \Omega_{i,d})_t$ and $\Omega_{i,i} = 0$ (No airport has an edge to itself). Example: During spring break, the edges connected to Miami Airport (MIA) are should remain constant for at least a week. If MIA were to have large delays due either to high traffic or some other exogenous factor, then flights to MIA from other airports have a high chance of being delayed as MIA struggles to get planes off the tarmac.

When the fused lasso estimates the precision matrices, the multiplier λ_2 controls how much change between precision matrices is tolerated. Note that $\boldsymbol{\beta}_i(t) \equiv \underline{\Omega_i}_t$.

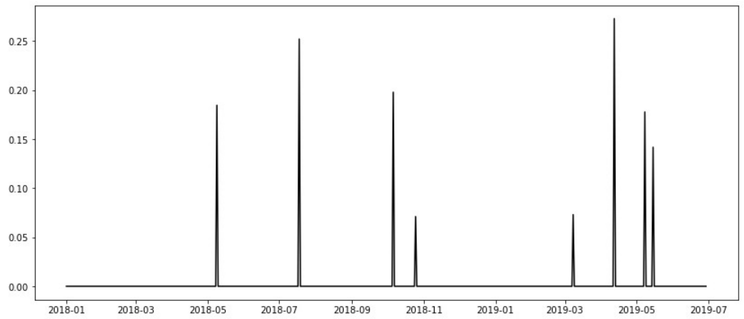
$$\underset{\{\hat{\boldsymbol{\beta}}_i(t)\}_{t=1}^T}{\arg\min} \left\{ \dots + \lambda_2 \sum_{t=2}^{T} \|\boldsymbol{\beta}_i(t) - \boldsymbol{\beta}_i(t-1)\|_1 \right\} \tag{8}$$

Large values of λ_2 force $\boldsymbol{\beta}_i(t)$ to be constant for all values of t; when λ_2 is small the edges vary wildly. To see an example, refer to the appendix. Theoretically, if a multivariate time series does in fact have breakpoints which signify structural changes between the variables, then there exists a $\hat{\lambda}_2$ that estimates a set of precision matrices that contain the seasonal dependencies between airports. What is of interest is the set of times in which $\underline{\Omega_i}_{\Delta t} \neq \Omega_0$. In other words, breakpoints are dates for which the structure of the graph changes from one day to the next.

Let $\{\hat{\Omega}_{\Delta t}\}_{t=2}^T$ be the set of sequential differences between estimated precision matrices. We can measure how much the structure of the network changes between each time step by taking the grand sum, or scalar sum of entries, of $\hat{\Omega}_{\Delta t}$. Note that because precision matrices are symmetric, it is sufficient to calculate $\{\delta_t\}_{i=2}^T$ such that

$$\delta_t = \sum_{i=1}^{d} \sum_{j<i}^{d} |\Omega_{t(i,j)}| \tag{9}$$

It turns out that as λ_2 increases, the noise due to large variations in the estimated network from one day to the next decreases, revealing the dates in which only significant changes in the network structure occur. In the case of American airlines when $\lambda_1 = 0.05$ and $\lambda_2 = 7$, this algorithm detected 5 dates for which there are significant structural changes in the network. Refer to the appendix to see the a progression of plots in which I incrementally increase the multiplier to see how the breakpoints are revealed as λ_2 increases.



BP	2018-05-09	2018-07-18	2018-10-06	2019-03-08	2019-04-12
δ_t	0.18	0.25	0.19	0.07	0.27

3.3 Prediction

The fundamental assumption of my predictive model is that the joint distribution of delay times $\underline{Y}'_t = (X_{t,1}, \dots, X_{t,10})'$ is invariant of yearly shifts. Stated plainly, if the network structure is known for an interval defined between two breakpoints, then the network structure a year in the future should be the same. This assumption is reasonable in so far as that the airline industry is driven by seasonal consumer demand. It is to be expected that exogenous factors add noise to the observed average daily delays, but the consumer driven interests are still there. Under these assumptions, I fit a piece wise 10 dimensional vector auto regressive model with lag = 1 to disjoint time intervals in 2018 and make predictions for the third quarter of 2019.

Recall that $\underline{Y}'_t = (X_{t,1}, \dots, X_{t,10})'$ is the vector of observations for the 10 airports of interest at time t. A VAR(1) model with coefficient matrix $\mathbf{A} \in \mathcal{M}_{\mathbf{dxd}}$ and white noise $\underline{\epsilon}_t \in \mathcal{R}^d$ is defined as the system of linear equations

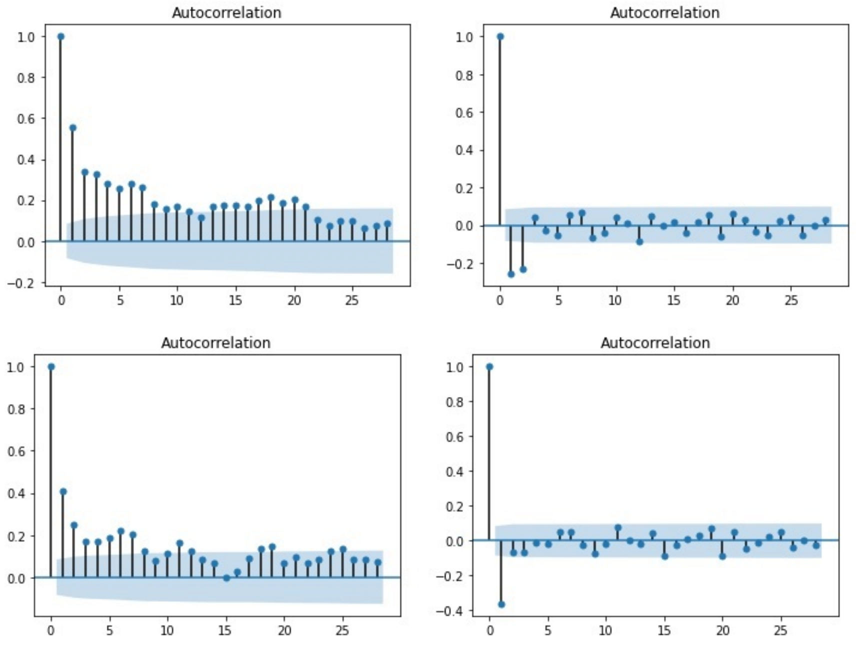
$$\underline{Y}_t = \mathbf{A}\underline{\mathbf{Y}}_{\mathbf{t-1}} + \underline{\epsilon}_{\mathbf{t}} \tag{10}$$

Let $\{\tau_i\}_{i=1}^N$ be the set of identified breakpoints such that $2019/07/01 \leq \tau_i \leq 2019/10/01$. A piece wise VAR(1) model for N segments can be expressed with an indicator function $\mathbb{1}_t$ that returns 1 when $\tau_i \leq t \leq \tau_{i+1}$ and 0 otherwise. Note that I ensured that no sequential breakpoints are less than 15 days apart, this protects against having to estimate pieces of the model on small sample sizes. Each coefficient matrix $\mathbf{A_i}$ is estimated on the ith segment of the 2018 data using the well known python package statsmodels.

In the case of American Airlines, the identified breakpoints which satisfy the conditions above are just outside the bounds of the interval that we want to estimate. As a result, there is only one interval on which we need to estimate $\mathbf{A}$; implying that the piece wise VAR(1) model reduces to the form of Equation 10. This is not the case for every airline company but in general breakpoints do tend lay around the dates listed in the table above.

$$\underline{Y}_t = \big(\mathbf{A_1}\mathbb{1}_\mathbf{t} + \dots + \mathbf{A_N}\mathbb{1}_\mathbf{t}\big)\underline{\mathbf{Y}}_{\mathbf{t-1}} + \underline{\epsilon_\mathbf{t}} \tag{11}$$

A key assumption of vector auto regression is that the time series is stationary. I apply first differencing to the log transformed average daily delay times in order make the series stationary. No longer does the model predict in units of minute, the series now represents the percent change in minutes between $t-1$ to t. Below are auto correlation plots for 2 of the 10 time series before and after differencing.



3.4 Results

After making the predictions I calculate MSE and conduct a Granger causality hypothesis test to find out which airports influence each other. Stated in terms of the data, high delays at airport i will cause high delays at airport j, thus inducing a direction between the two. Recall that the Gaussian graphical models discussed in section two are undirected graphs. Essentially, airport i is said to 'Granger-cause' airport j if predictions of j based on it's own delay time series, and on the time series for i, make better predictions than if only previous observations from j were used to predict itself. For each airport i, I test the null hypothesis that all other airports do not provide any predictive power for i. The python package statsmodels has this functionality. Below is a table of the mean squared errors for the predicted time series for American Airlines.

AA	DFW	ORD	PHX	CLT	MIA	DCA	LAX	PHL	LGA	BOS
MSE	326.1	803.0	220.4	227.6	193.7	865.0	83.8	334.4	618.8	426.6
p-value	0.864	0.004	0.005	0.135	0.176	0.065	0.001	0.427	0.306	0.141

Take note of the airport with the smallest and largest p-value, LAX and DFW. LAX not only has the small p-value but the smallest MSE. This is in line with the definition of of Granger causality. In the case of DFW, Dallas/Fort Worth International airport in the geographc network in section has the greatest number of edges to other airports. One might guess that DFW's delays are dependent on the collection of airports it shares an edge with. The Granger causality hypothesis test would say this is wrong, we fail to reject that DFW's delays are not influenced by the other airports. Not only that, but it's p-value is the largest. This gives us good reason to believe that DFW is the reason why other airports experience delays, not vice versa.

4. Conclusion

Faults: see how the graph structure changes from year to year given a fixed interval, more refined data and higher lag

In this work I developed the probabilistic motivation for analyzing flight delays in terms of a graphical model and introduced the fused lasso as a method for time series segmentation. Under assumption that the multivariate joint distribution of the graphical model is invariant to yearly shifts, I predicted flight delays in the third quarter of 2019 using a piece wise VAR(1) model. Lastly, I discuss the predictions and conduct hypothesis tests which enable direction to be introduced into our undirected graphical model with 95% confidence.

The discovered relationships between airports through the VAR(1) model can be used as a starting point of further research. Consider a black swan event that causes airport X_i to have extreme delays on a given day. This information is critical to airline companies as it can allow them to optimally reroute flights; ultimately preventing cascading delays through other airports, saving resources, and increasing customer satisfaction.

References

Note that I did not have time to properly cite my references. Hopefully links will suffice.

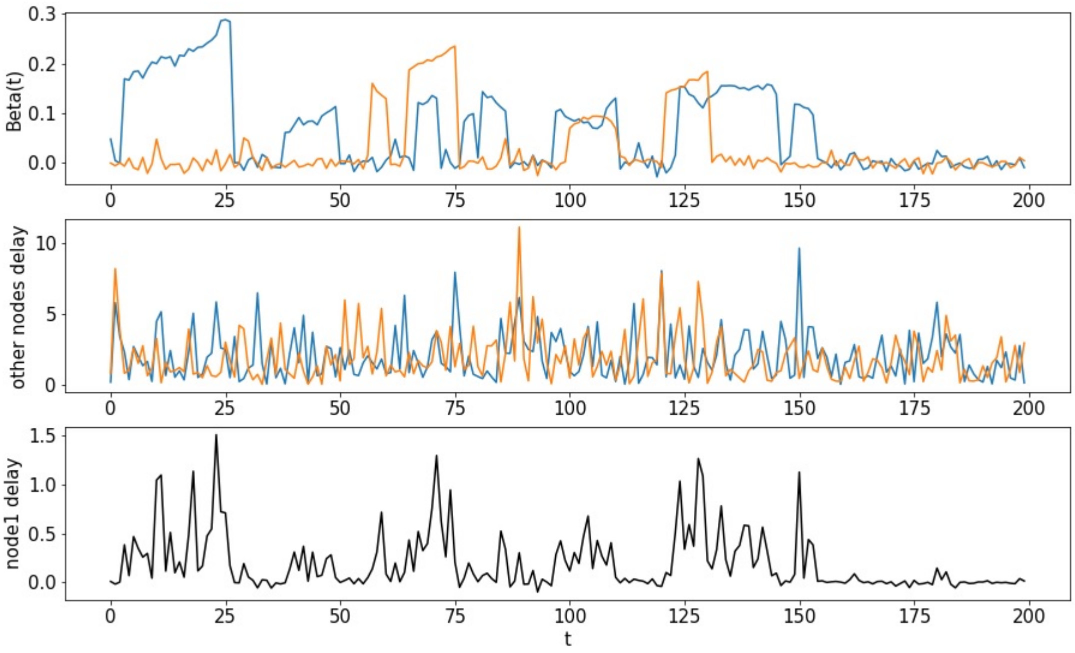
- https://arxiv.org/pdf/math/0608017.pdf
- https://web.stanford.edu/group/SOL/papers/fused-lasso-JRSSB.pdf
- http://www.cs.cmu.edu/ epxing/Class/10708-16/note/10708_scribe_lecture10.pdf
- http://swoh.web.engr.illinois.edu/courses/IE598/handout/markov.pdf
- http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.766.8454rep=rep1type=pdf
- https://faculty.washington.edu/ezivot/econ584/notes/varModels.pdf
- https://arxiv.org/pdf/1912.07761.pdf
- https://www.youtube.com/watch?v=_vQ0W_qXMxk
- https://www.researchgate.net/profile/Nooshin_Omranian/Segmentation_of_biological_multivariate_time-series_data
- https://arxiv.org/pdf/1311.4175.pdf
- https://aspmhelp.faa.gov/index.php/Types_of_Delay

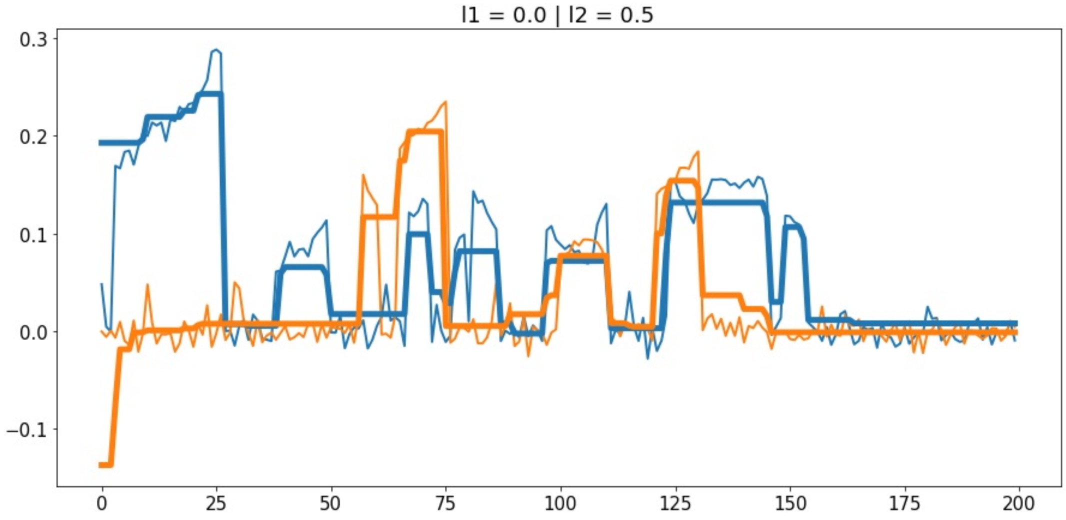
Appendix

Simulation study

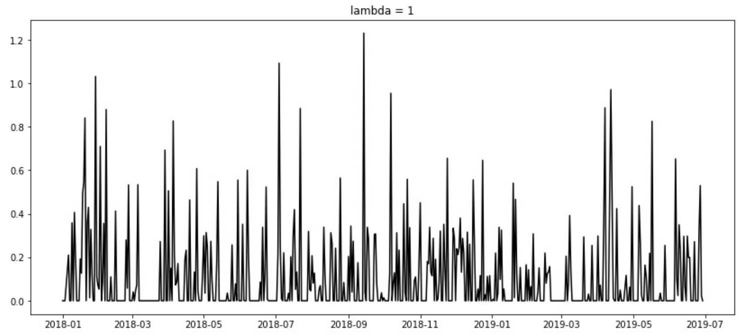
In order to make an artificial time series **y** for a hypothetical airport upon which to test the fuse lasso, I simulate two other time series that **y** is directly dependent on. First I generate two sequences of coefficients that are approximately zero a majority of the time but with random kicks to significant changes. These are the population parameters used to generate the series **y**. Let $e_t \sim \mathcal{N}(0, 0.01)$.

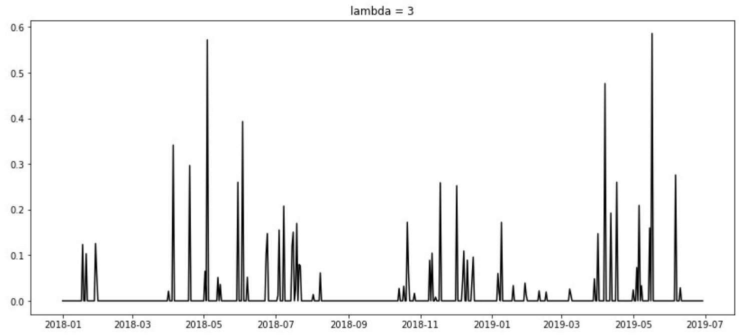
$$y_t = \big(x_1(t), x_2(t)\big) \begin{pmatrix} \beta_1(t) \\ \beta_2(t) \end{pmatrix} + e_t \tag{12}$$

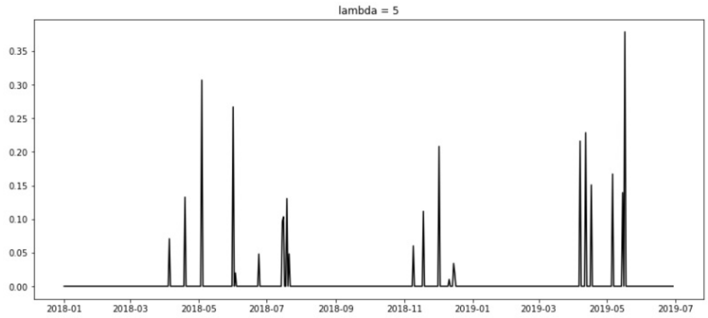




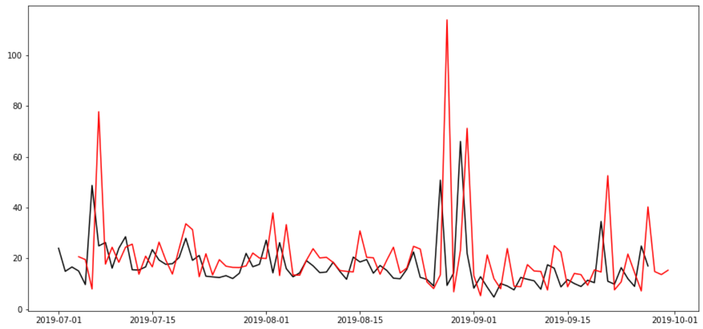
4.1 Behavior of λ_2



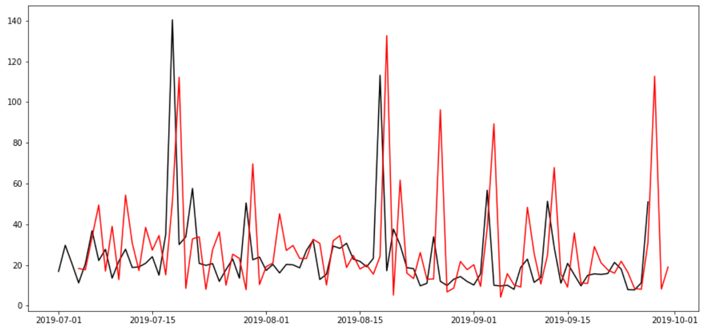




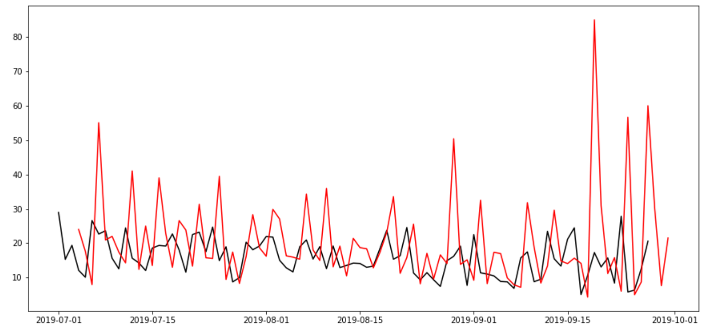
Prediction Plots



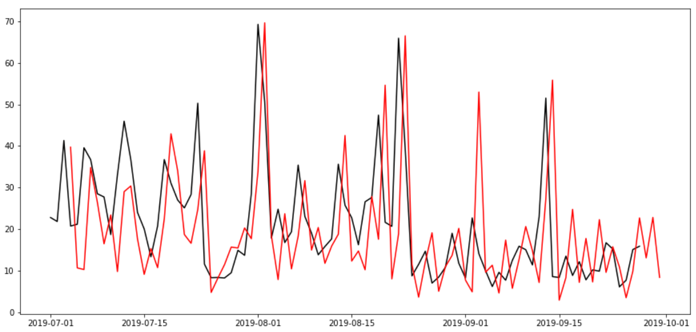
DFW



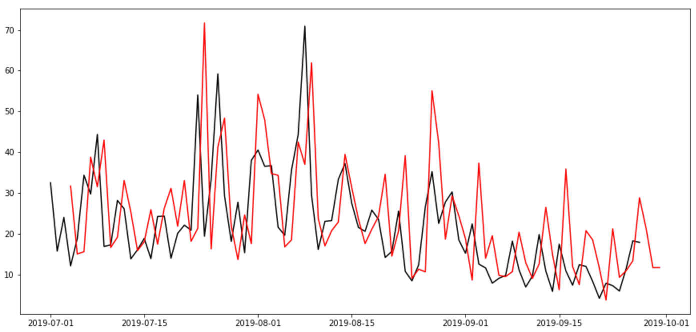
ORD



PHX

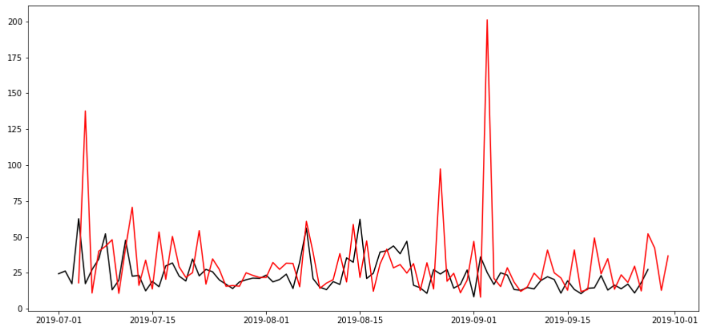


CLT

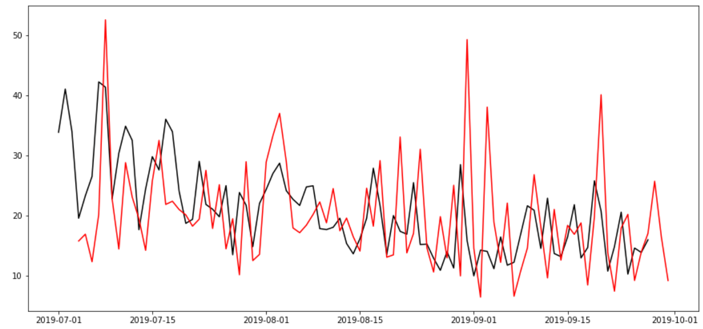


MIA

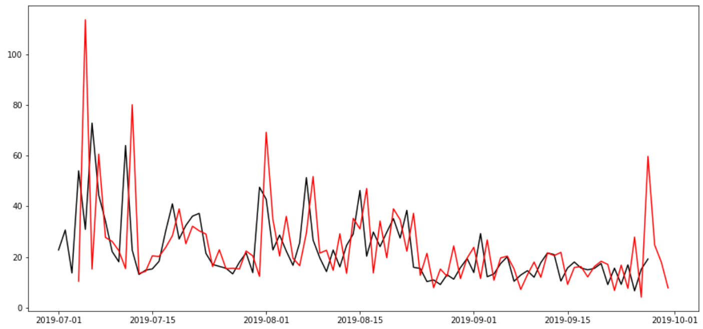
Estimated networks for other airline companies



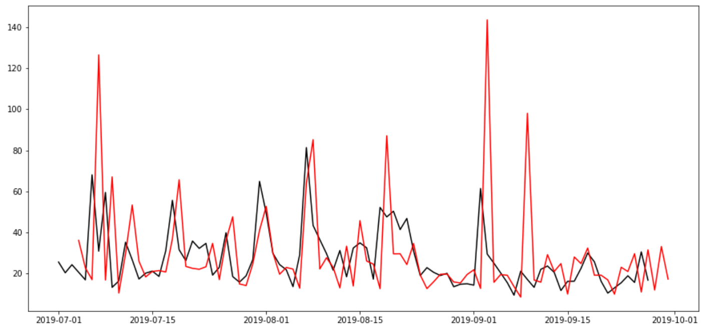
DCA



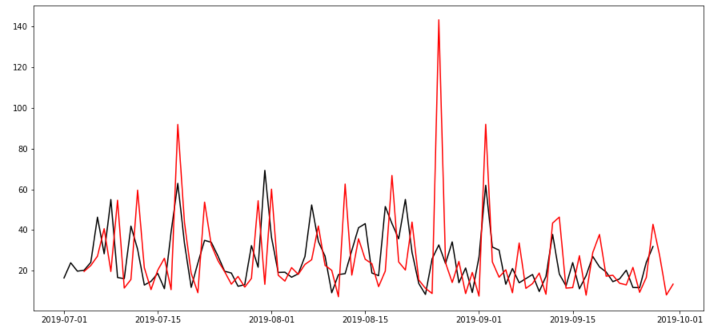
LAX



PHL



LGA


140
120
100
80
60
40
20
2019-07-01
2019-07-15
2019-08-01
2019-08-15
2019-09-01
2019-09-15
2019-10-01

BOS