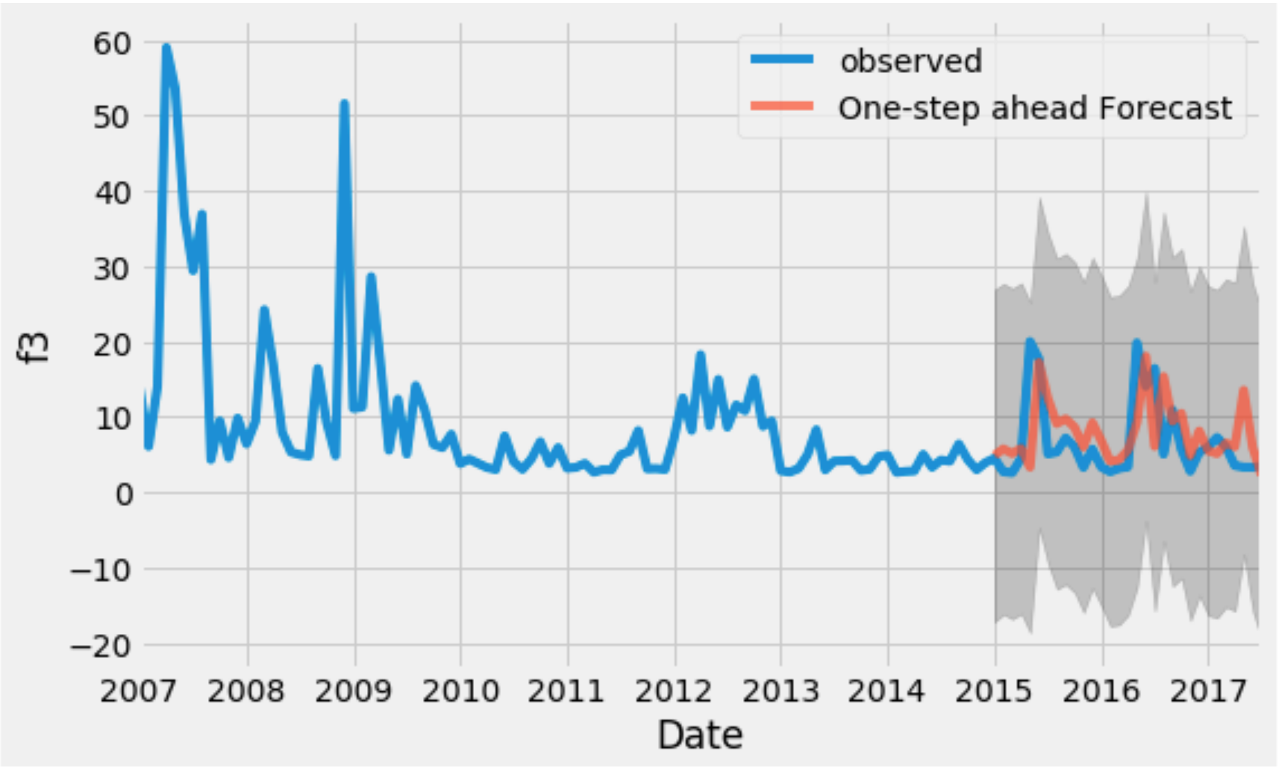
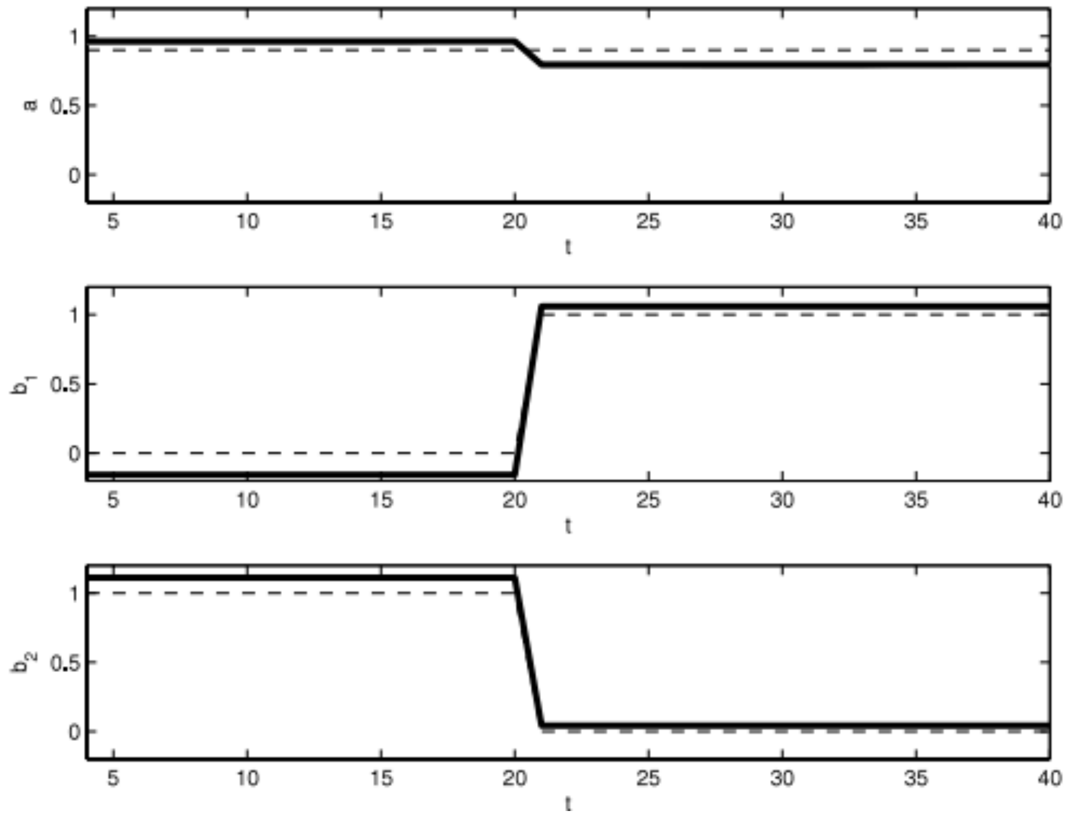
TS1 Progress Report

Code Available [HERE]



Time Series Prediction



Time-Variant Analyses

N. Fazeli, A. Issaoui, P. Lerchi, M. Parent, S. Wei

April 26th, 2018

Our Method

1. **Select Optimal Features**
   - Simple, intuitive analysis
   - Aligns with other team
2. **Predict Feature Time Series**
   - Create stationary features
   - Reasonable initial result
3. **Linear Regression Baseline**
   - Basic code complete
4. **Validate on Real Price Data**
   - Mixed results
5. **Time Varying Weight Analysis**
   - Stanford paper implementation
   - Useful tool for case studies

Project Goals

Goal One: Develop an initial framework for implementing time series prediction

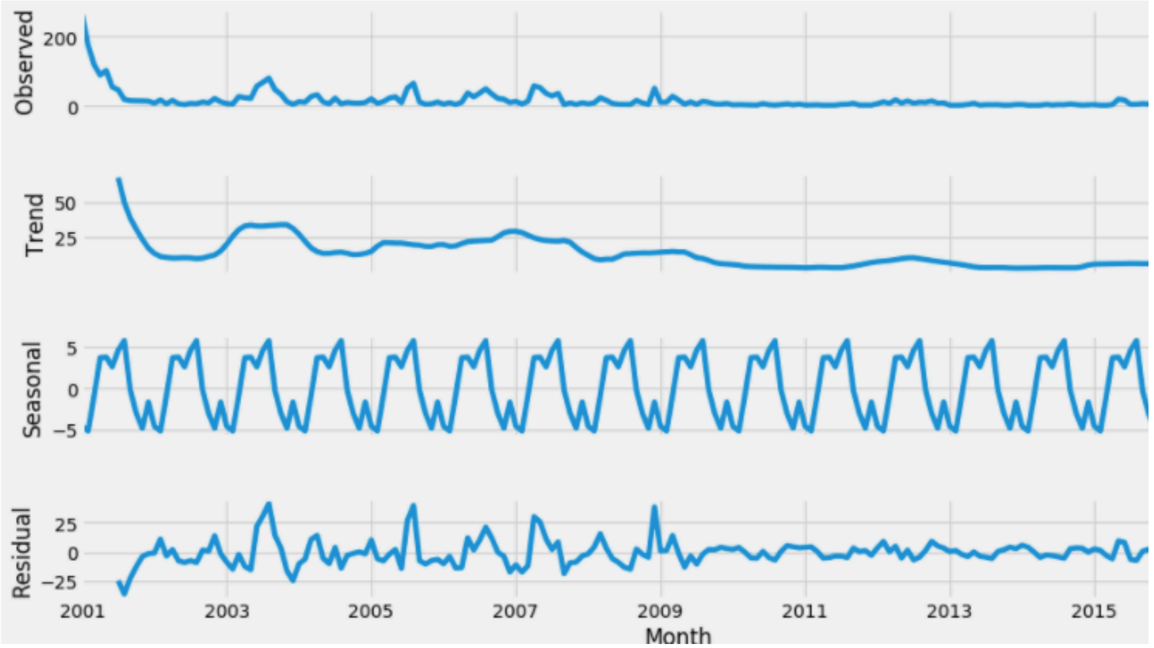
Goal Two: Help users gain insight on disruptive market changes through dynamic feature impact analysis

Feature Selection

1. **High Correlation with Price**
   - Understandable result
   - Verify with PGE employee

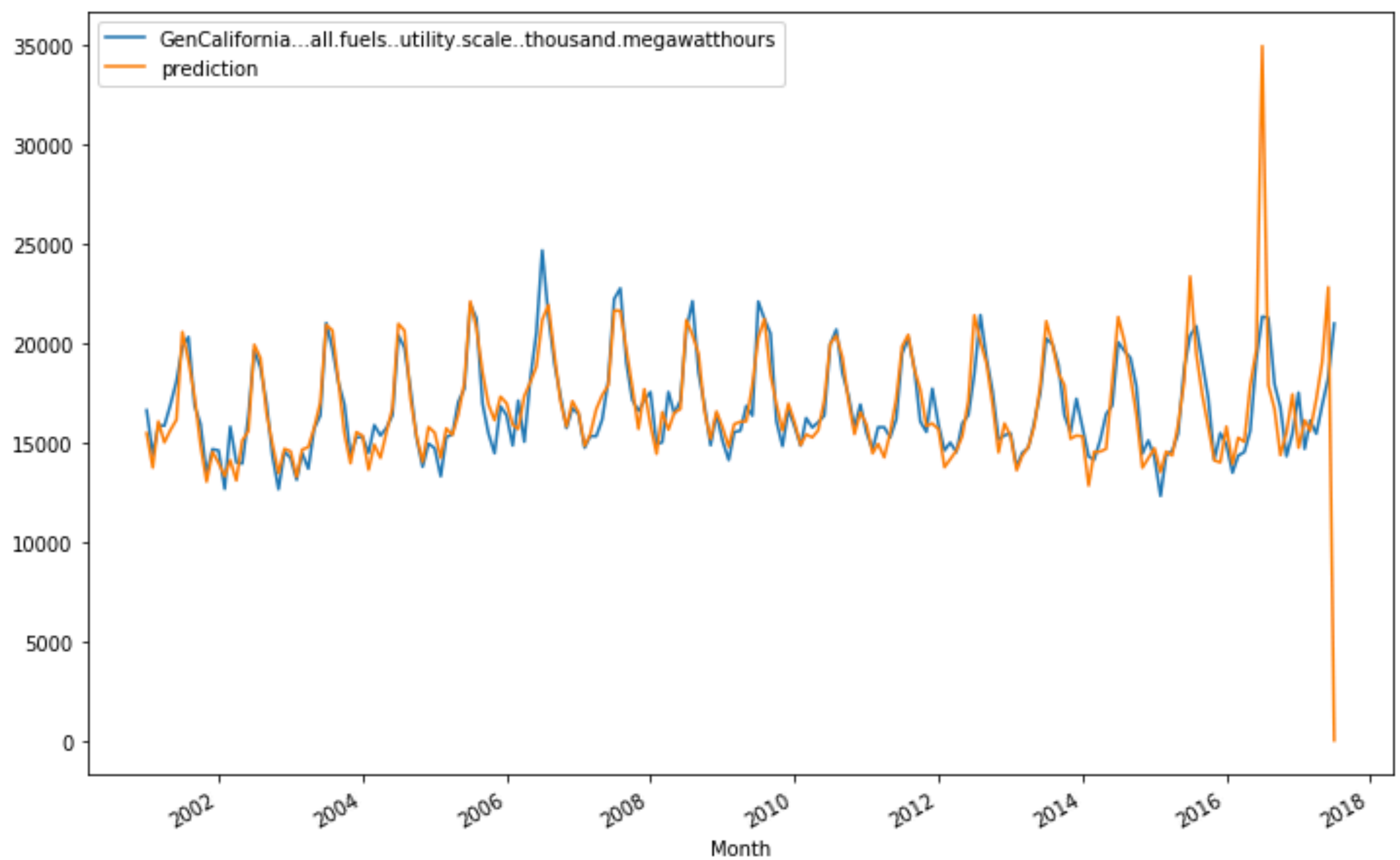
2. **Make Stationary**
   - Use Dickey-Fuller test
   - First difference is sufficient

3. **Decompose Seasonality**

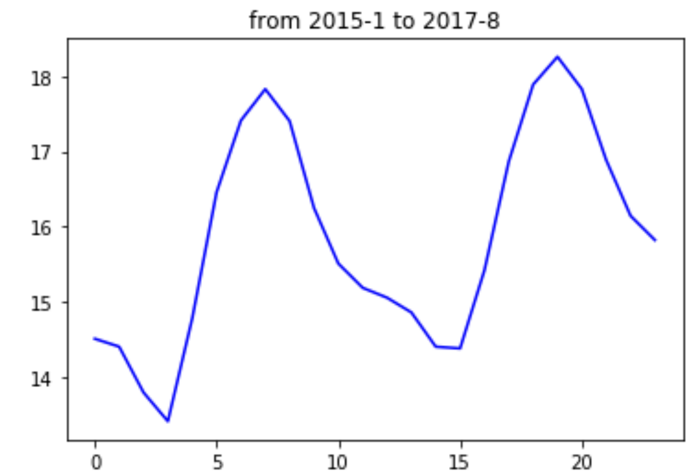


	Simplified	Original Defination
0	Y	PrCalifornia...all.sectors.cents.per.kilowatthour
1	f3	GenCalifornia...petroleum.liquids.thousand.meg...
2	f5	GenCalifornia...natural.gas.thousand.megawatth...
3	f6	GenCalifornia...other.gases.thousand.megawatth...
4	f13	GenCalifornia..Geothermal.electric.utility
5	f15	GDP
6	f16	Av.Temp
7	f18	Av.Rel.Humid
8	f22	Hydro.Consumption.TrillBtu.
9	f26	WoodConsump..TrillBTU.
10	f37	Percentage.of.Industrial.Sector.Consumption.fo...
11	f38	Natural.Gas.Price..Electric.Power.Sector..Doll...
12	f48	Electricity.Retail.Sales..Total..Billion.Kilow...
13	f49	Electricity.Direct.Use..Billion.Kilowatthours.
14	f60	Natural.Gas.Consumed.by.the.Other.Industrial.S...
15	f70	Retail.sales.of.electricity.monthly.California...

Time Series



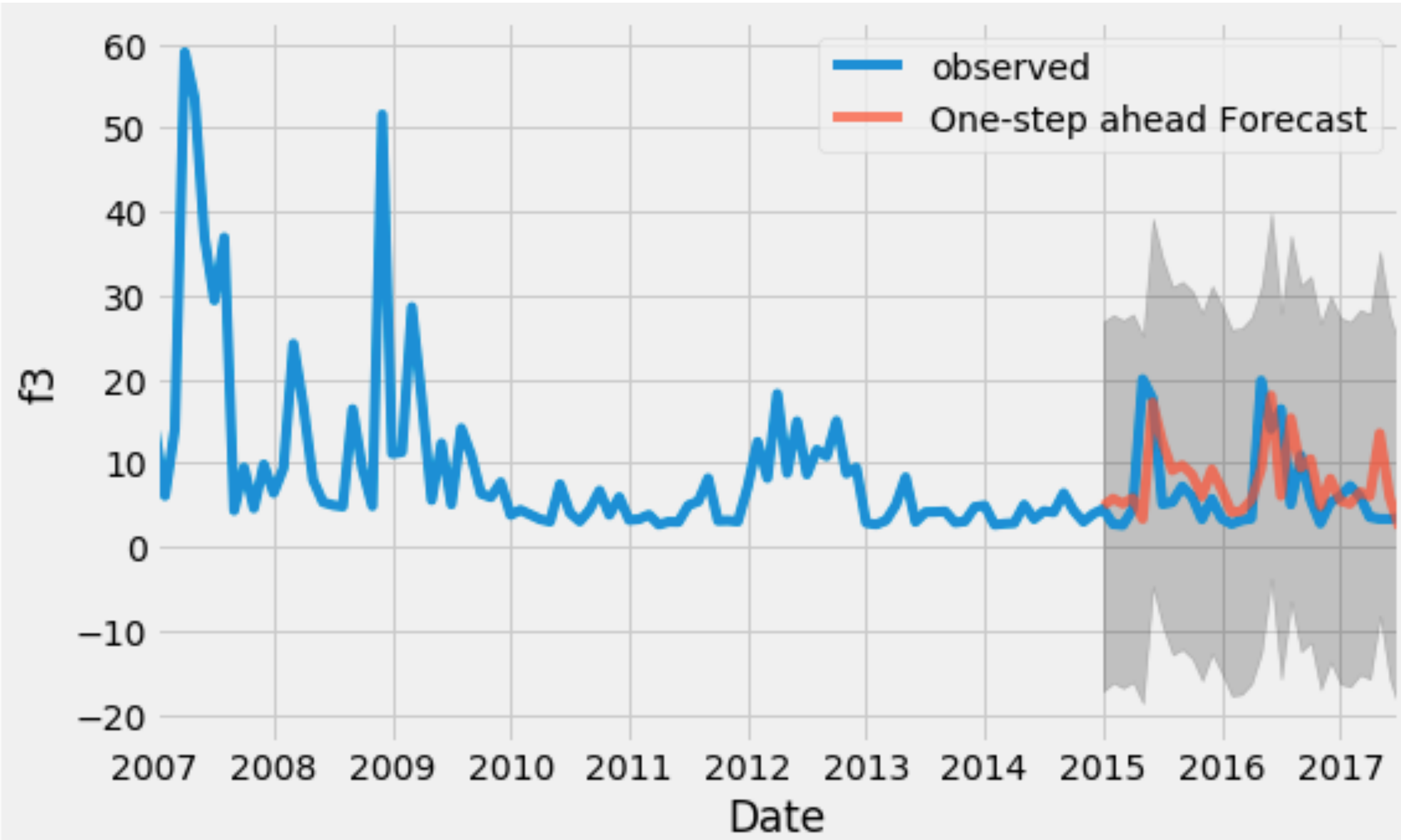
	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.6272	0.029	21.754	0.000	0.571	0.684
ma.L1	-2.0183	0.060	-33.567	0.000	-2.136	-1.900
ma.L2	1.0229	0.061	16.682	0.000	0.903	1.143
ma.S.L12	-1.6419	7.377	-0.223	0.824	-16.101	12.817
ma.S.L24	0.6435	4.721	0.136	0.892	-8.609	9.896
sigma2	108.5474	800.735	0.136	0.892	-1460.865	1677.960




```
# Error
error
```

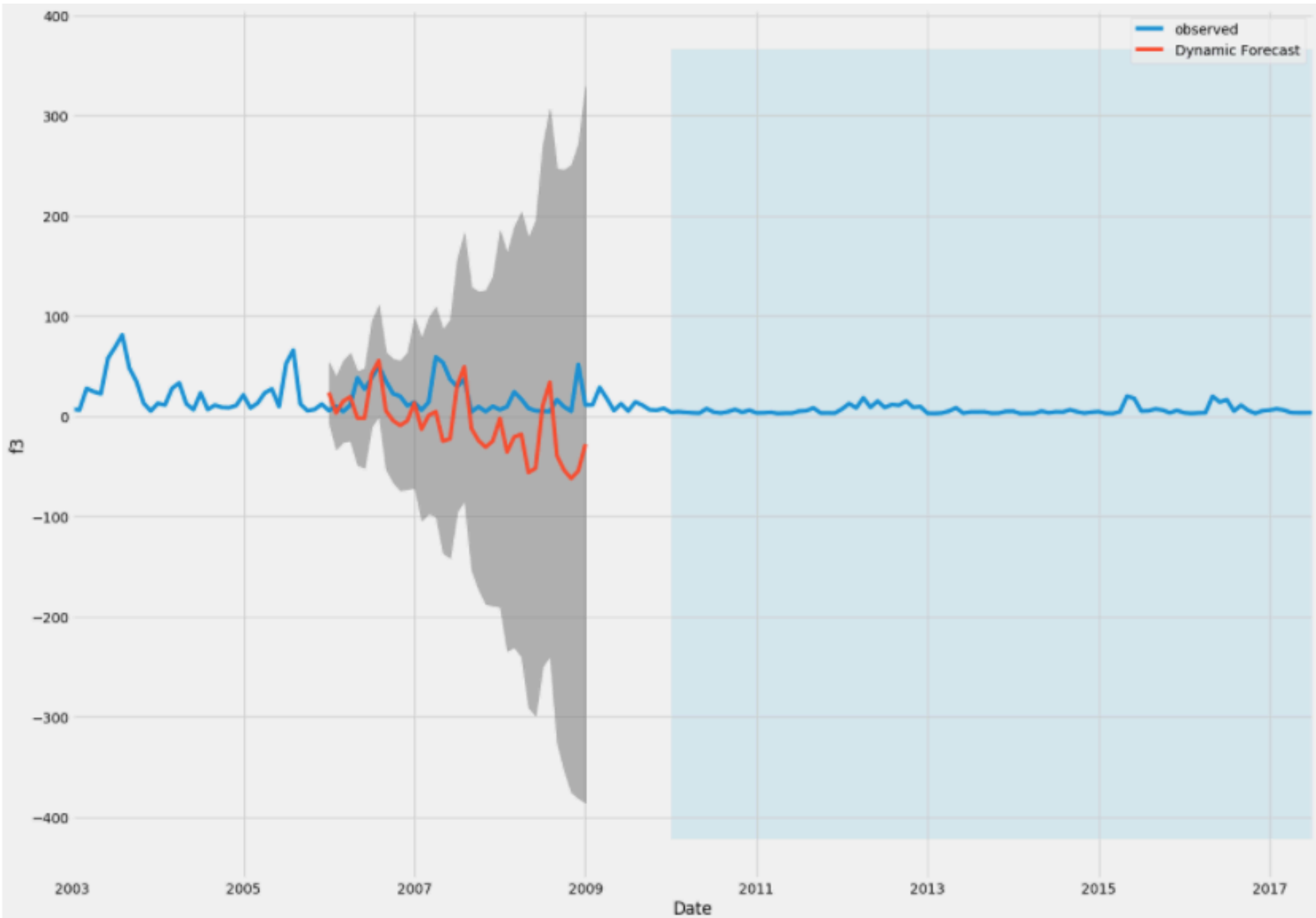
```
0.8024830223442727
predicted=14.506427, expected=14.550000
predicted=14.401120, expected=14.300000
predicted=13.791988, expected=14.220000
predicted=13.408586, expected=12.640000
predicted=14.775977, expected=14.760000
predicted=16.455376, expected=16.120000
predicted=17.409834, expected=17.310000
predicted=17.827590, expected=17.190000
predicted=17.402254, expected=17.360000
predicted=16.247365, expected=15.620000
predicted=15.506575, expected=15.030000
predicted=15.184271, expected=14.410000
predicted=15.052558, expected=14.740000
predicted=14.856494, expected=14.530000
predicted=14.400784, expected=14.340000
predicted=14.377486, expected=12.670000
predicted=15.417520, expected=14.940000
predicted=16.865276, expected=16.240000
predicted=17.888186, expected=16.930000
predicted=18.257737, expected=17.150000
predicted=17.827553, expected=16.680000
predicted=16.883228, expected=14.180000
predicted=16.142433, expected=14.990000
predicted=15.819268, expected=14.880000
```

Prediction



observed
One-step ahead Forecast
60
50
40
30
20
10
0
−10
−20
f3
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
Date

Dynamic Forecast?



Linear Regression

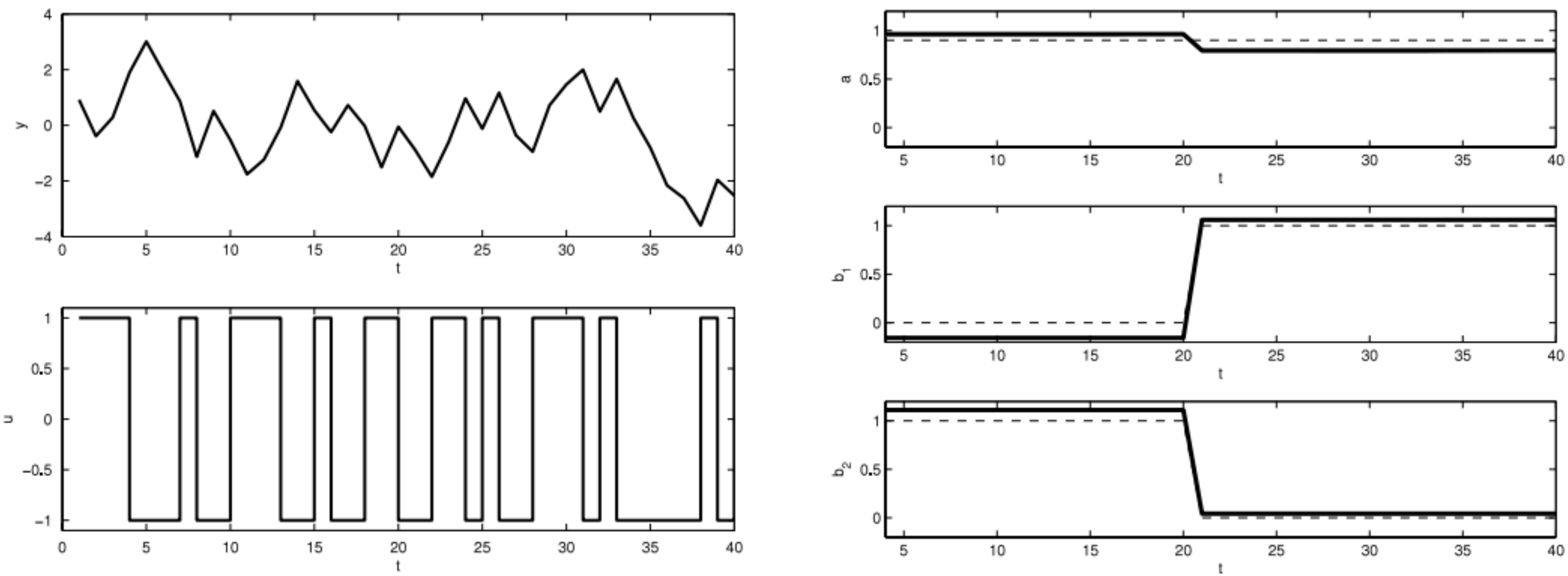
```
# Linear Regression Model
from sklearn.linear_model import LinearRegression
linreg = LinearRegression()
linreg.fit(X_train, y_train)
print('Intercept: \n', linreg.intercept_)
print('Coefficients: \n', linreg.coef_)
```

```
Intercept:
 17.286327564830035
Coefficients:
 [ 2.66753464e+01 -3.30918520e+01 -2.66514194e+01  3.30789746e+01
  4.59779903e-01  3.19790609e-03 -9.70528617e-02 -3.21269468e-03
  6.43840211e+00 -6.43739008e+00 -1.52808447e-02  1.22266573e-02
  7.57754171e-02  7.57754171e-02 -4.68676657e-02  4.65264535e-02
 -1.87590994e+00  3.16047966e-03  1.69754023e-03  3.99995699e-03
 -1.97815416e-02 -1.34609875e-01]
```

```
MSE, Training Data: 0.05
MSE, Test Data: 16708.30
```

Time Varying Weight

$$\min_{\theta(t)} \sum_{t=1}^{N} \|y(t) - \varphi^T(t)\theta(t)\|^2 + \lambda \sum_{t=2}^{N} w(t)\|\theta(t) - \theta(t-1)\|_{\mathrm{reg}}$$



See Stanford Paper for additional detail

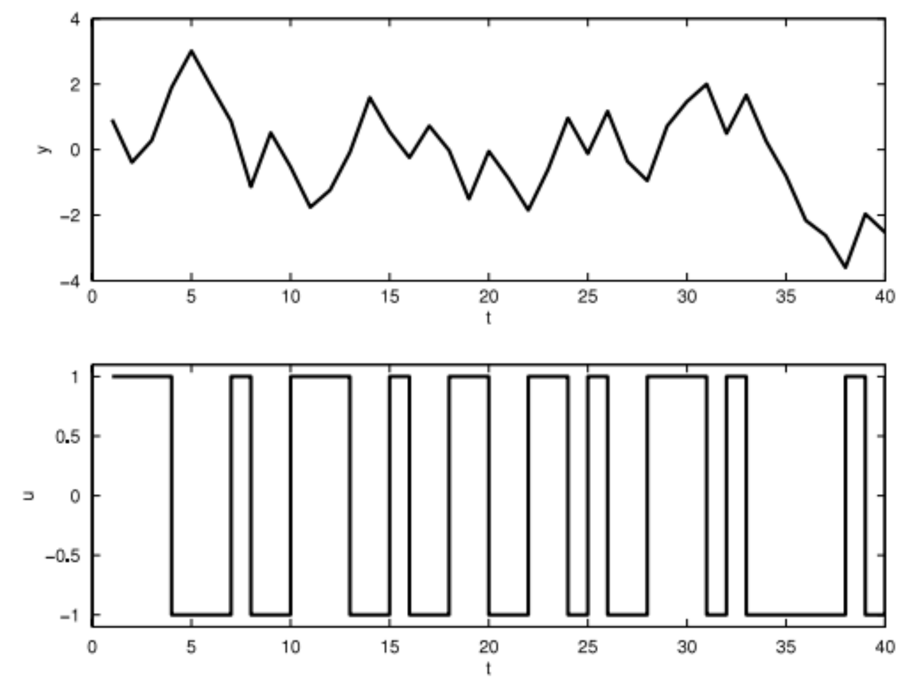
Impact

1. **More realistic modeling**
   - Feature significance varies over time
   - Results are intuitively displayed

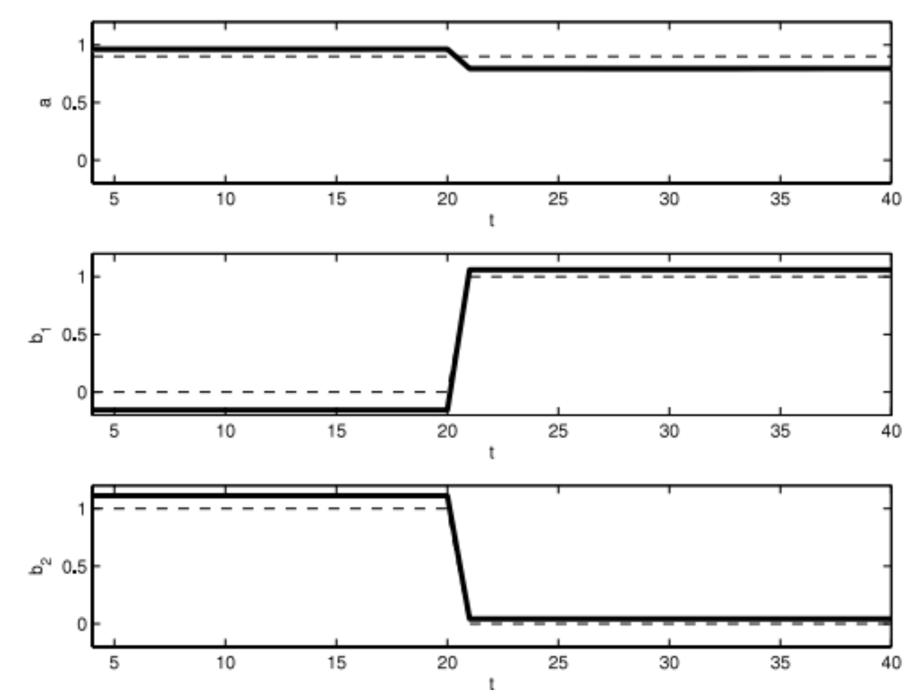
2. **Simplify complex feature analysis**
   - Algorithm auto-selects dominant features
   - Consider a larger scope of variables

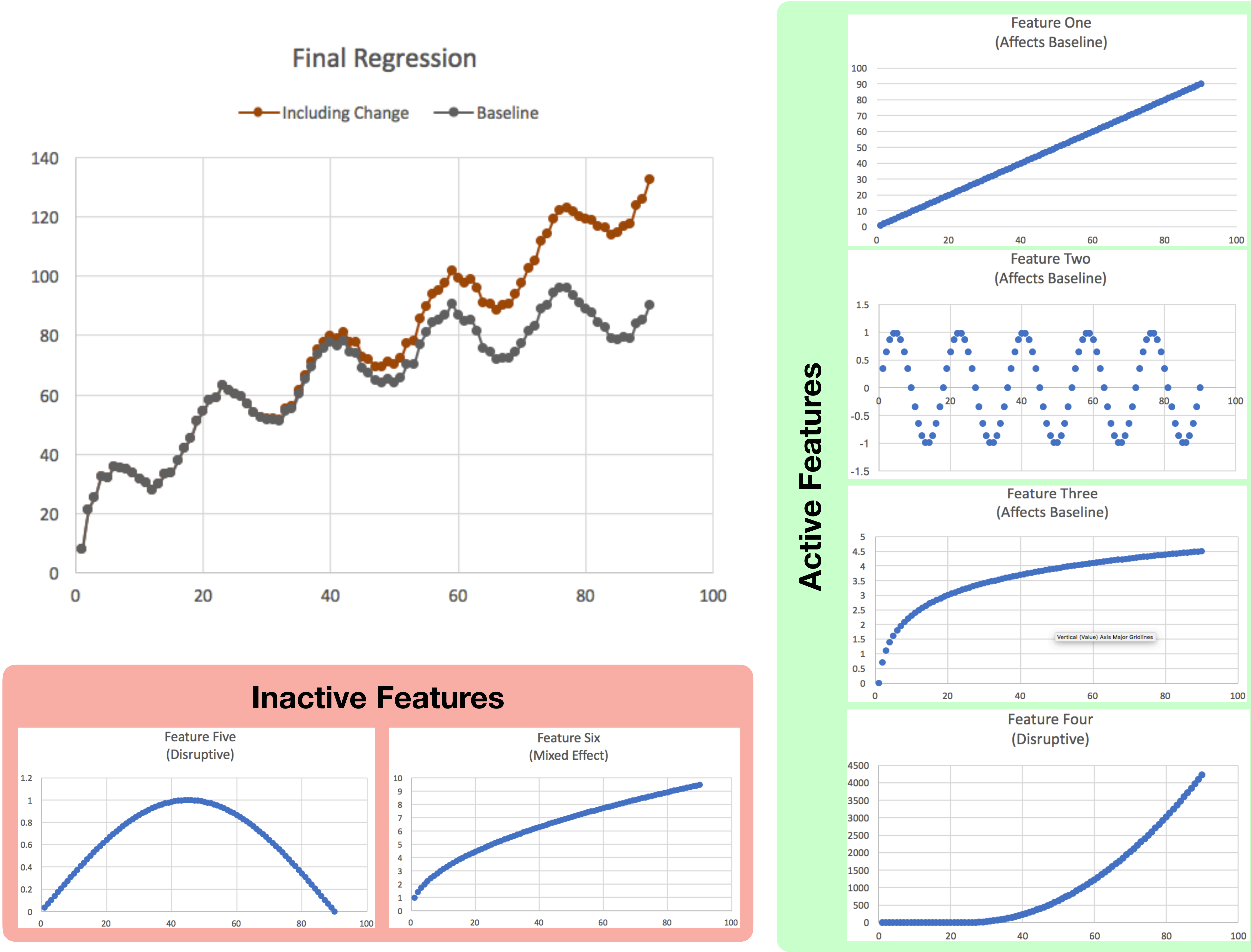
3. **Clarify impact of disruption**
   - Separate baseline variables from disruptors
   - Proportional effect of irregular behavior

4. **Qualify future predictions**
   - Case studies
   - Parametric sweep
   - Confidence intervals



$$y(t) + ay(t-1) = b_1u(t-1) + b_2u(t-2)$$




Final Regression
Including Change
Baseline
Inactive Features
Feature Five
(Disruptive)
Feature Six
(Mixed Effect)
Active Features
Feature One
(Affects Baseline)
Feature Two
(Affects Baseline)
Feature Three
(Affects Baseline)
Vertical (Value) Axis Major Gridlines
Feature Four
(Disruptive)