Natural Resource Consumption Modeling

THINKFUL CAPSTONE

Kara Grosse

Research Questions

- How has natural gas consumption changed over time?
  - What external factors impact natural gas consumption over time
- Can we predict future natural gas consumption with reasonable accuracy?

Product Concept

Provide a tool for natural resource extractors and utilities to predict demand for different natural resource so they tailor their extraction efforts accordingly

Steps

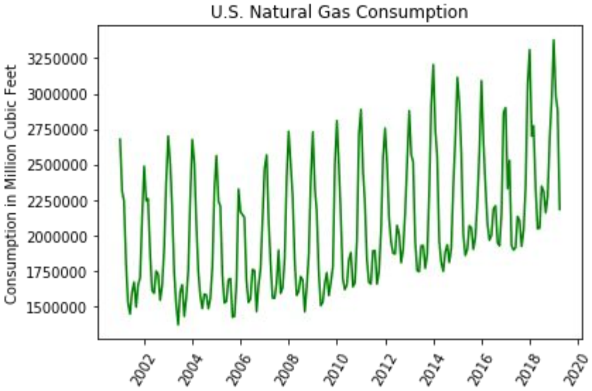
1. Data gathering via an api
2. Data cleaning
3. Exploratory Analysis
4. Model Prep
5. Model Building
6. Forecasting
7. Conclusions

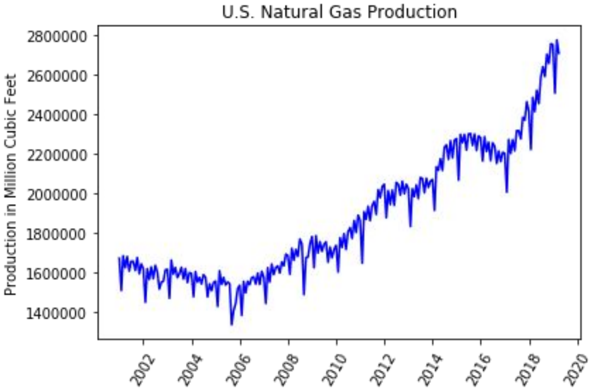
Data Gathering

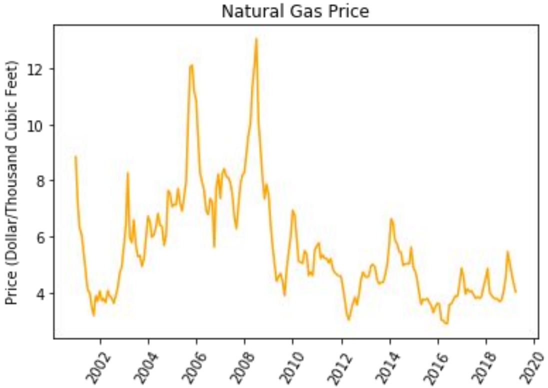
- The Energy Information Administration (EIA) provides open source data through its api
- Currently, EIA's API contains the following main data sets:
  - Hourly electricity operating data, including actual and forecast demand, net generation, and the power flowing between electric systems
  - 408,000 electricity series organized into 29,000 categories
  - 30,000 State Energy Data System series organized into 600 categories
  - 92,836 International energy series
  - Natural resource categories:
    - 115,052 petroleum series and associated categories
    - 34,790 U.S. crude imports series and associated categories
    - 11,989 natural gas series and associated categories
    - 132,331 coal series and associated categories

eia Independent Statistics & Analysis
U.S. Energy Information Administration

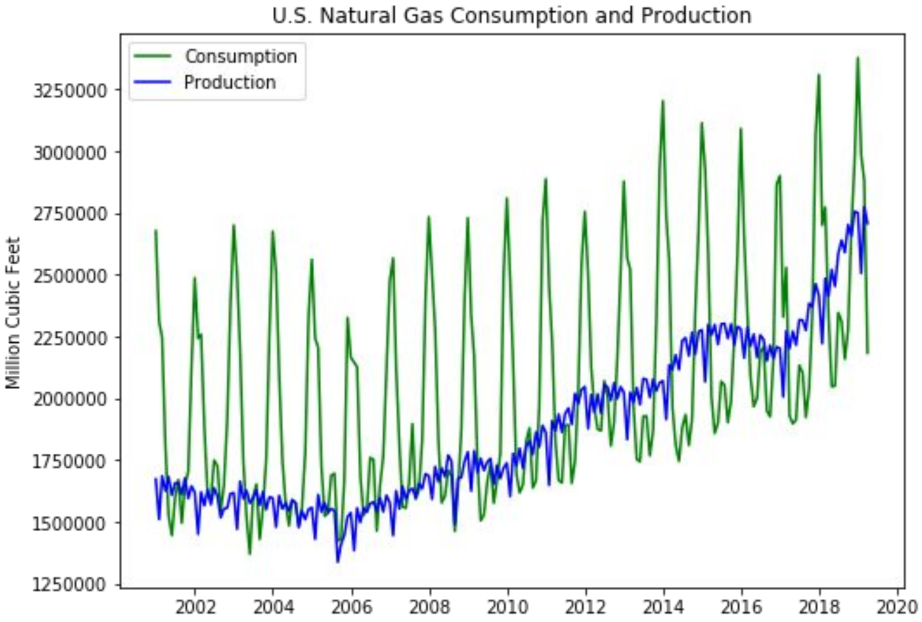
Time Series Exploration

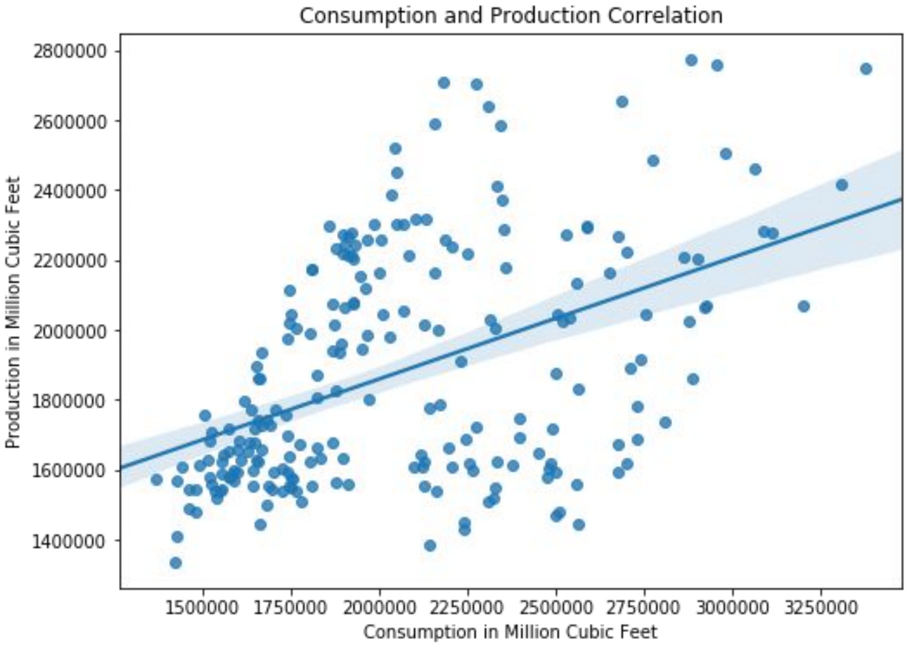




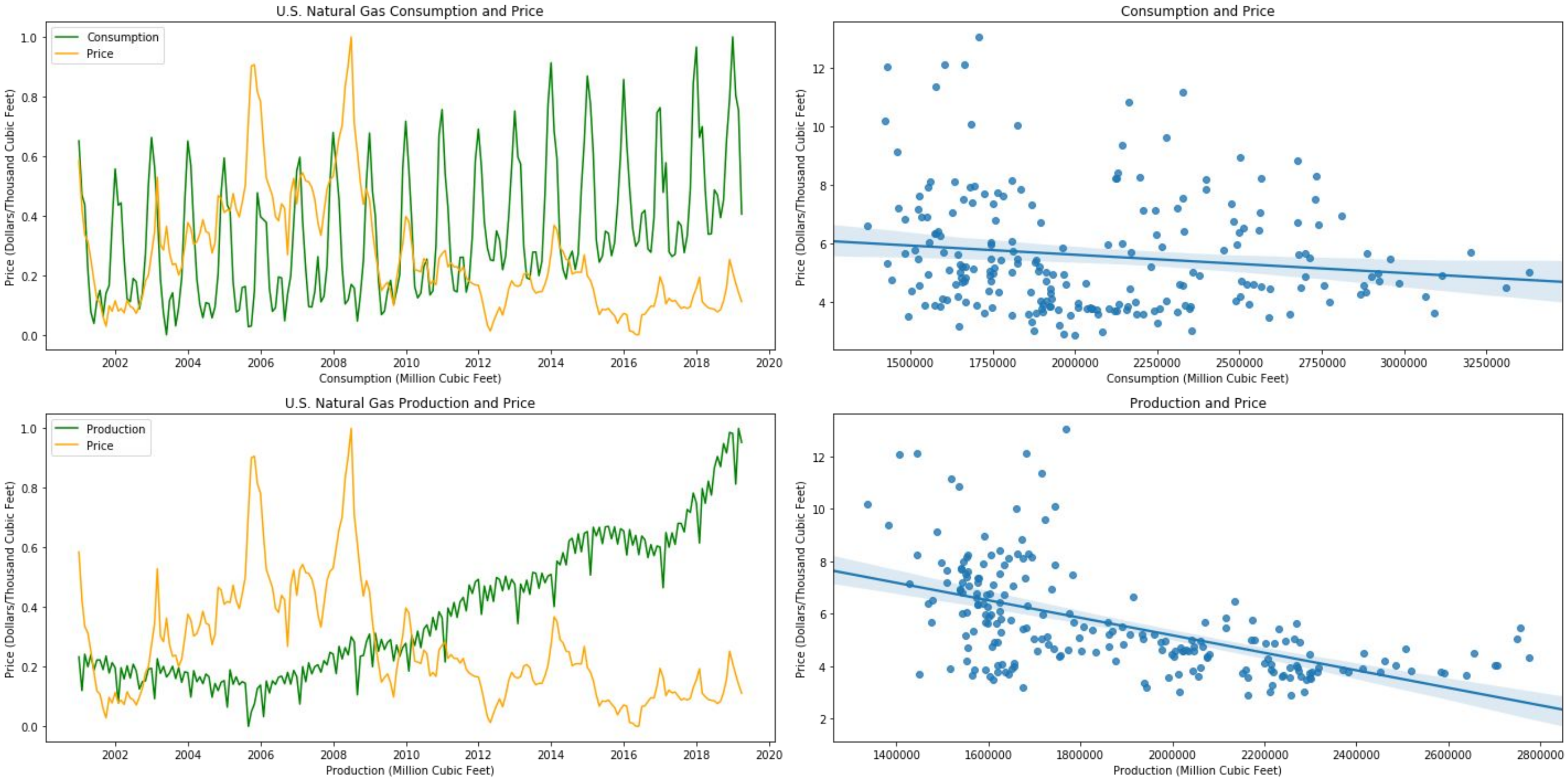


Time Series Exploration





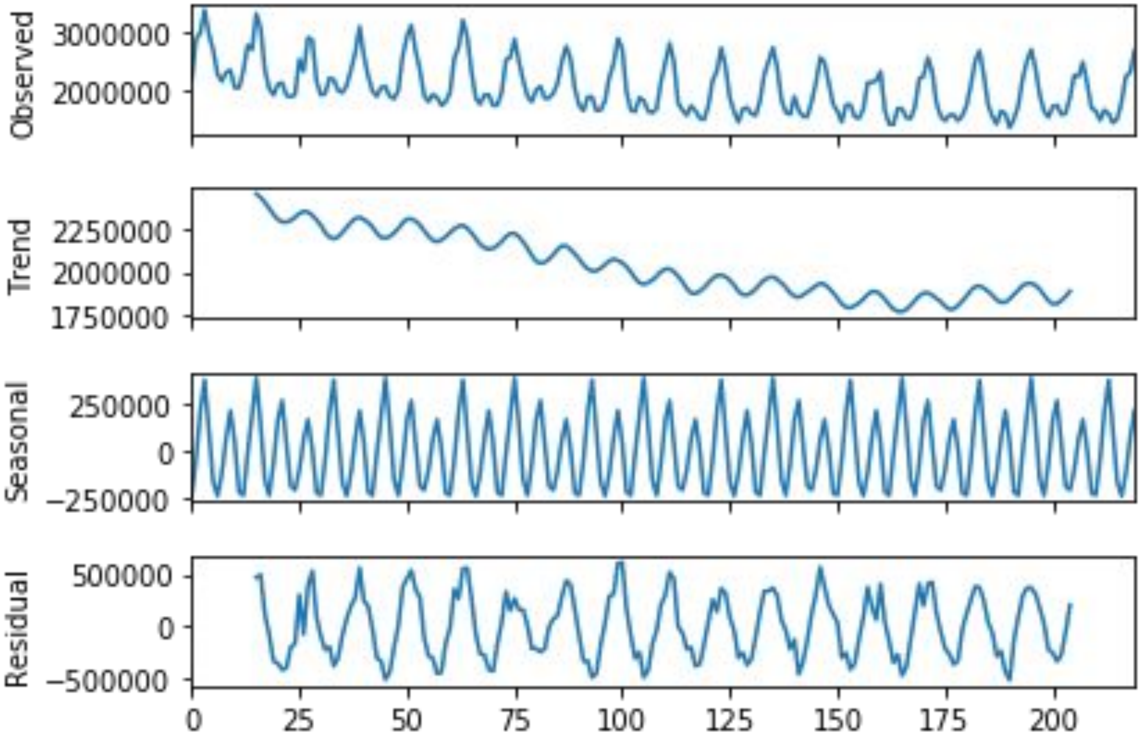
Consumption/Production Regression: LinregressResult(slope=0.34730665694477514, intercept=1163567.4029101448, rvalue=0.46210672357551386, pvalue=4.887378587666091e-13, stderr=0.045141934716189845)



Consumption/Price Regression: LinregressResult(slope=-6.258746508874706e-07, intercept=6.870731926624818, rvalue=-0.1404466034813413, pvalue=0.03737707183948859, stderr=2.9882834046382836e-07)

Production/Price Regression: LinregressResult(slope=-3.3364986223876108e-06, intercept=11.850250921510362, rvalue=-0.5627113158772621, pvalue=9.000967164134613e-20, stderr=3.3197131011480474e-07)

Time Series Decomposition



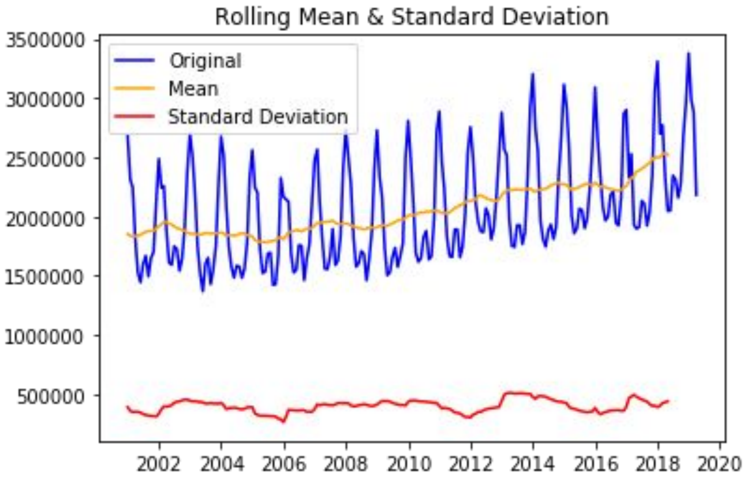
The decomposition shows a negative trend, seasonality, and high variability for the consumption data

ARIMA

- AutoRegressive Moving Average Model
- Three parameters: p,d,q
    - p: Trend autoregression order
    - d: Trend difference order
    - q: Trend moving average order
- Good for short term forecasting with limited data
- Good for data that does not have a lot of variability

ARIMA Model Prep

- Make sure data is stationary
  - Rolling statistics
  - Adfuller test



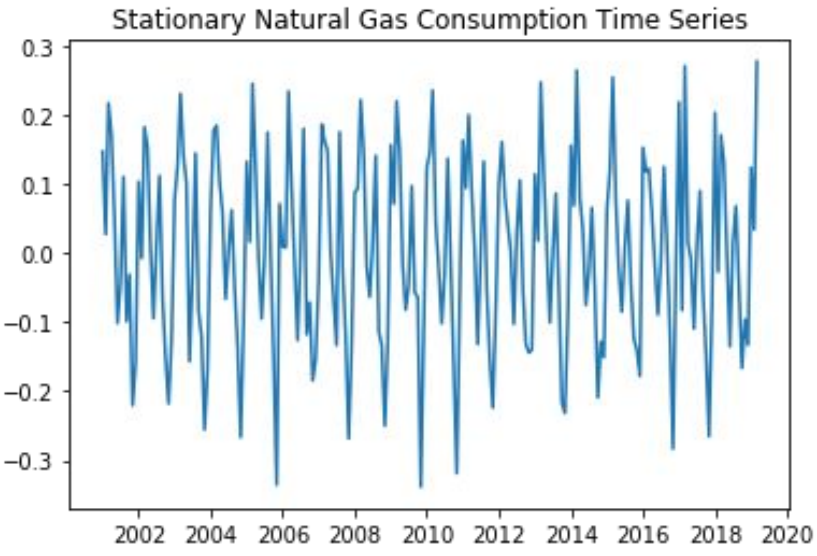
In [22]:
```
# Write function to test for stationarity
def stationarity_test(name, x):
    result = adfuller(x)
    print(name,':')
    print('ADF Statistic %f' % result[0])
    print('p-value: %f' % result[1])
    if result[1] > 0.05:
        print('{} data set is NOT stationary, differencing required!\n'.format(name))
    else:
        print('{} is stationary, hooray!\n'.format(name))

stationarity_test('Consumption', df.Cons_Mcf)
```

```
Consumption :
ADF Statistic -2.166910
p-value: 0.218474
Consumption data set is NOT stationary, differencing required!
```

ARIMA Model Prep

- Log the data to get smaller values
- Make sure differenced data is stationary
- Graph differenced data

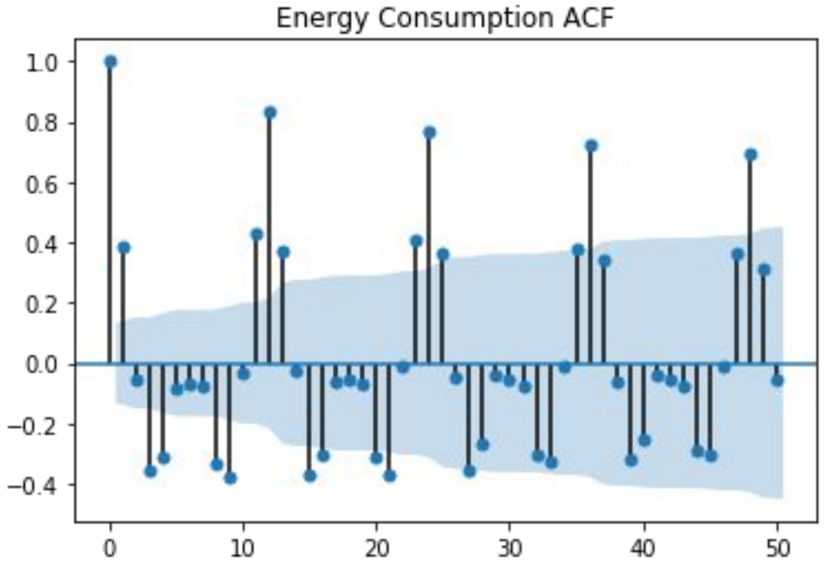


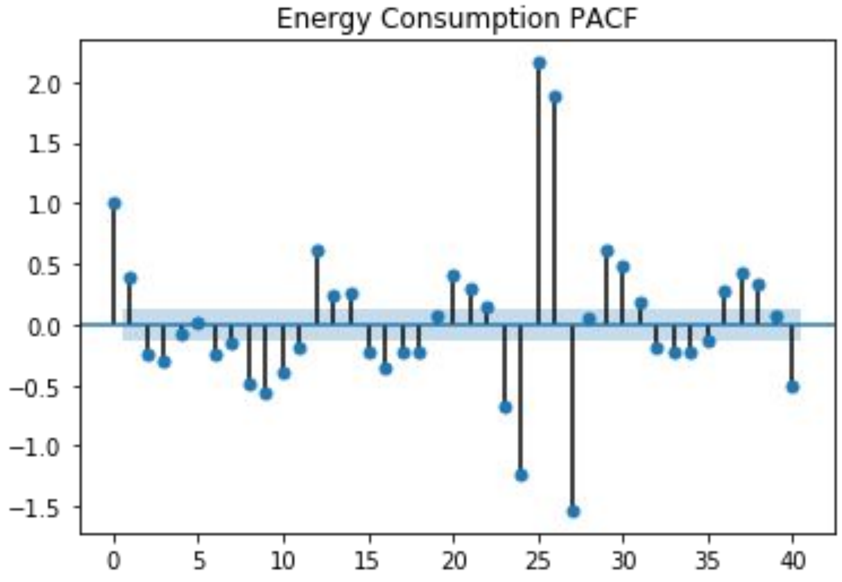

```
[27]: #make sure the differenced data is stationary
stationarity_test('Consumption', df.diff_1[1:])

Consumption :
ADF Statistic -4.836713
p-value: 0.000046
Consumption is stationary, hooray!
```

ARIMA Model Prep

- Construct ACF and PACF graphs to obtain number ranges for autoregressive and moving average parameters





ARIMA Model Building

- Create function to generate predictions for an ARIMA model with a given p,d,q order
- Create a function to determine the best p,d,q order by minimizing the mean squared error based on ranges of values for each variable gathered from the autocorrelation analysis

```
ARIMA config: (0, 0, 0); MSE:0.016726444026987048
ARIMA config: (0, 0, 1); MSE:0.015239618606061546
ARIMA config: (0, 0, 2); MSE:0.015441480215509739
ARIMA config: (0, 0, 3); MSE:0.024357696996484438
ARIMA config: (0, 0, 4); MSE:0.020748287639253953
ARIMA config: (0, 0, 5); MSE:0.020975025459198004
ARIMA config: (0, 0, 6); MSE:0.021372602830970915
ARIMA config: (0, 0, 7); MSE:0.026260807621902246
ARIMA config: (0, 1, 0); MSE:0.04312829839745987
ARIMA config: (0, 1, 1); MSE:0.036079532038552006
ARIMA config: (0, 1, 2); MSE:0.015489242492055492
ARIMA config: (0, 1, 3); MSE:0.015697695075550334
ARIMA config: (1, 0, 0); MSE:0.015415631649293604
ARIMA config: (1, 0, 1); MSE:0.019801906342879882
ARIMA config: (1, 1, 0); MSE:0.03848431301879281
ARIMA config: (2, 0, 0); MSE:0.01550105384572352
ARIMA config: (2, 0, 2); MSE:0.034828725314735645
ARIMA config: (2, 0, 5); MSE:0.027677939851917122
ARIMA config: (2, 1, 0); MSE:0.04146348031526297
ARIMA config: (3, 0, 0); MSE:0.01767647897586606
ARIMA config: (3, 0, 2); MSE:0.03171251797521869
ARIMA config: (3, 1, 0); MSE:0.04199281613855394
ARIMA config: (4, 0, 0); MSE:0.018152705308979285
ARIMA config: (4, 0, 2); MSE:0.028045298961478126
ARIMA config: (4, 0, 4); MSE:0.034793705211120435
ARIMA config: (4, 1, 0); MSE:0.03636920253439426
ARIMA config: (4, 1, 4); MSE:0.02378216810146974
ARIMA config: (5, 0, 0); MSE:0.01792595682925318
ARIMA config: (5, 0, 2); MSE:0.03979419877729804
ARIMA config: (5, 0, 4); MSE:0.041342004985149276
ARIMA config: (5, 0, 5); MSE:0.04095415058857704
ARIMA config: (5, 1, 0); MSE:0.036333074178917436
Best ARIMA config: (0, 0, 1)
MSE: 0.015239618606061546
```

ARIMA Model Building

- With the best configuration, build the model and evaluate the model summary

ARMA Model Results

Dep. Variable:	diff_1	**No. Observations:**	196
Model:	ARMA(0, 1)	**Log Likelihood**	135.204
Method:	css-mle	**S.D. of innovations**	0.121
Date:	Mon, 22 Jul 2019	**AIC**	-264.408
Time:	11:12:57	**BIC**	-254.574
Sample:	04-01-2017	**HQIC**	-260.427
	- 01-01-2001		

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.0018	0.012	0.147	0.883	-0.022	0.026
ma.L1.diff_1	0.4334	0.062	7.026	0.000	0.313	0.554

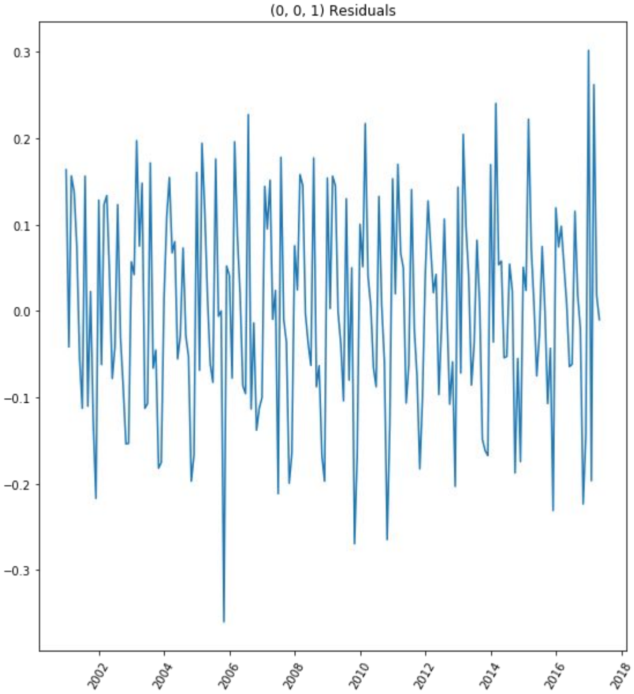
Roots

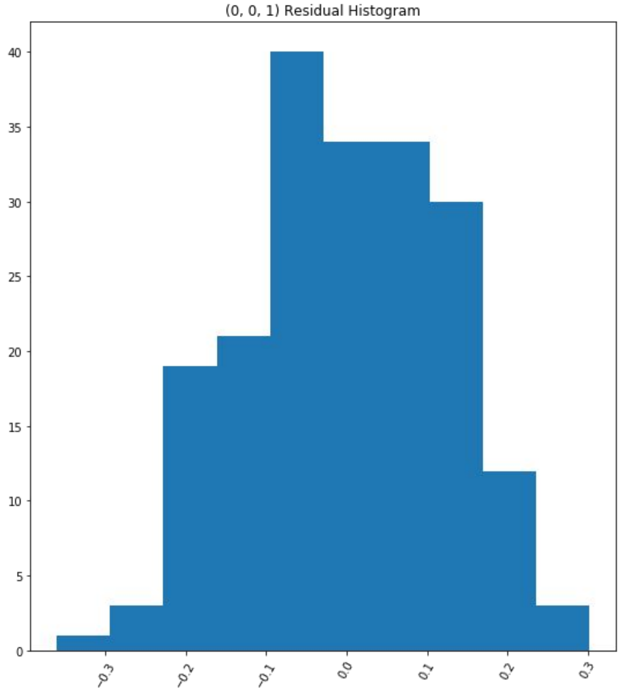
	Real	Imaginary	Modulus	Frequency
MA.1	-2.3071	+0.0000j	2.3071	0.5000

ARIMA Model Building

- Evaluate Residuals

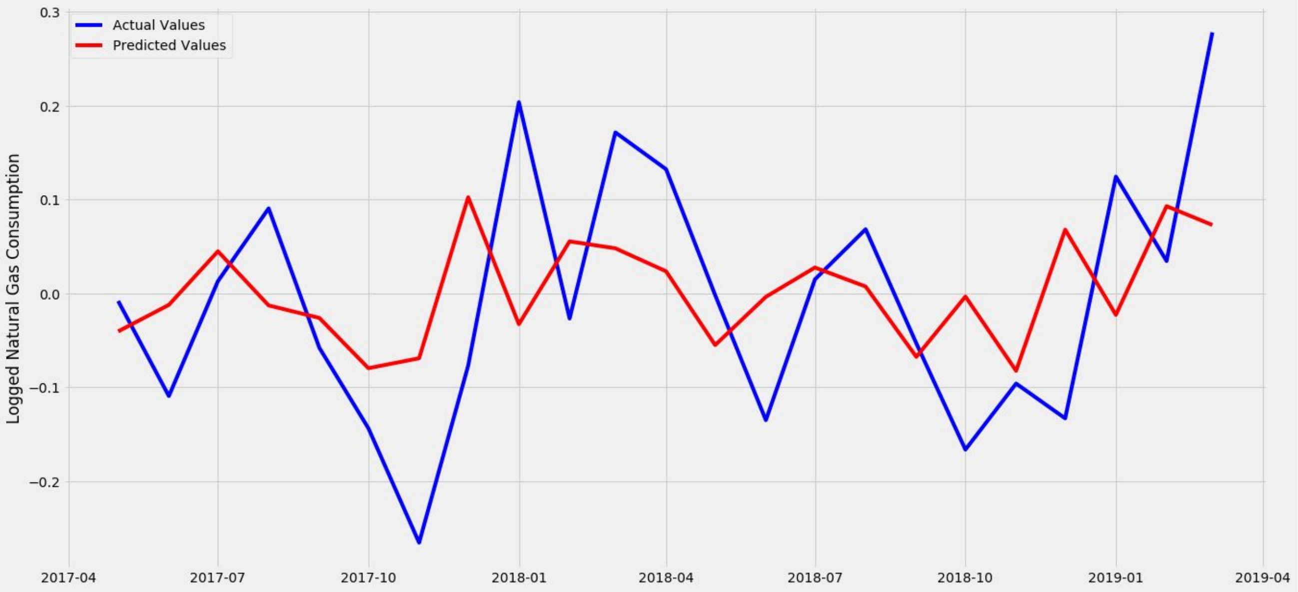
count	197.000000
mean	-0.000071
std	0.121381
min	-0.359942
25%	-0.082739
50%	-0.000261
75%	0.086160
max	0.301754





ARIMA Model Building

- Create predictions and compare against test data
- Calculate MSE

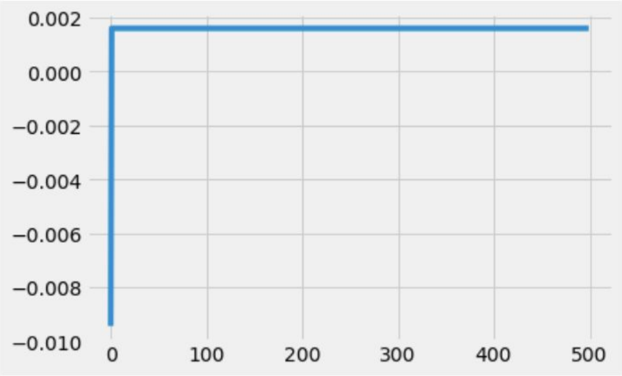



```
predicted=0.072930, expected=0.278024
predicted=0.092958, expected=0.034549
predicted=-0.022747, expected=0.124439
predicted=0.067938, expected=-0.132966
predicted=-0.082314, expected=-0.095905
predicted=-0.003251, expected=-0.166239
predicted=-0.067478, expected=-0.053121
predicted=0.007405, expected=0.068404
predicted=0.027620, expected=0.015047
predicted=-0.003592, expected=-0.134881
predicted=-0.054906, expected=-0.001746
predicted=0.023602, expected=0.132274
predicted=0.048080, expected=0.171505
predicted=0.055497, expected=-0.026694
predicted=-0.032694, expected=0.203725
predicted=0.102482, expected=-0.076769
predicted=-0.068950, expected=-0.265342
predicted=-0.079559, expected=-0.143509
predicted=-0.026013, expected=-0.057792
predicted=-0.012698, expected=0.090603
predicted=0.044892, expected=0.013128
predicted=-0.012151, expected=-0.109201
predicted=-0.040386, expected=-0.007975
Test 0,0,1 MSE: 0.015
```

ARIMA Model Building

- Forecasts with this model reverted to the mean extremely quickly

```
[-0.00942936  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605  0.00160605
```



SARIMAX Model Building

- Try an alternate ARIMA model to try to lower the MSE and produce viable forecasts
- SARIMAX: Seasonal ARIMA model with exogenous variables
    - Already accounts for differencing and trend via hyperparameters
    - Include p, d, q as well as P,D,Q,m
        - P: Seasonal autoregressive order.
        - D: Seasonal difference order.
        - Q: Seasonal moving average order.
        - m: The number of time steps for a single seasonal period.
    - An m of 12 is important for our model because we have a cycle that repeats annually, as we saw in the ACF graph

SARIMAX Model Building

- Evaluate with hold-out groups
  - Set up train and test data
    - Test = 2 years (24 months)
    - No differencing needed for this model
- Create a function to iterate through values of p,d,q and P,D, and Q with regards to m to find the model with the lowest AIC
  - Parameters for lowest AIC:
    - p,d,q = (1,1,1)
    - P,D,Q=(1,0,1)
    - m = 12

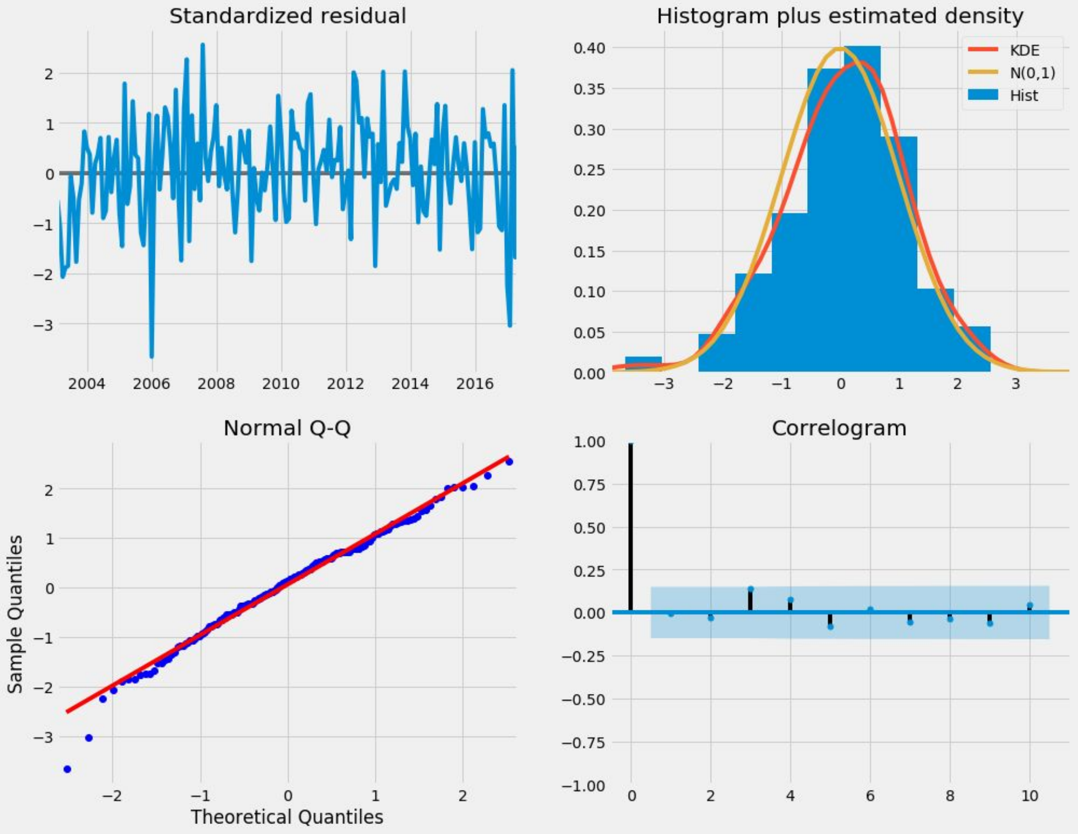
```
#try with non-differenced training data
sarimax_config(tra)
```

```
ARIMA(0, 0, 0)x(0, 0, 0, 12)12 - AIC:1410.057848299334
ARIMA(0, 0, 0)x(0, 0, 1, 12)12 - AIC:1314.7160344093231
ARIMA(0, 0, 0)x(0, 1, 0, 12)12 - AIC:-428.43313443469543
ARIMA(0, 0, 0)x(0, 1, 1, 12)12 - AIC:-391.4435329247656
ARIMA(0, 0, 0)x(1, 0, 0, 12)12 - AIC:-439.3292939637213
ARIMA(0, 0, 0)x(1, 0, 1, 12)12 - AIC:-460.202506494418
ARIMA(0, 0, 0)x(1, 1, 0, 12)12 - AIC:-393.12298717409385
ARIMA(0, 0, 0)x(1, 1, 1, 12)12 - AIC:-388.795092479335 4
ARIMA(0, 0, 1)x(0, 0, 0, 12)12 - AIC:1173.1302230247352
ARIMA(0, 0, 1)x(0, 0, 1, 12)12 - AIC:1133.5373036725869
ARIMA(0, 0, 1)x(0, 1, 0, 12)12 - AIC:-453.42341684634334
ARIMA(0, 0, 1)x(0, 1, 1, 12)12 - AIC:-427.07129179889364
ARIMA(0, 0, 1)x(1, 0, 0, 12)12 - AIC:-435.7855245334674
ARIMA(0, 0, 1)x(1, 0, 1, 12)12 - AIC:-412.0172901320668
ARIMA(0, 0, 1)x(1, 1, 0, 12)12 - AIC:-425.9657092269746
ARIMA(0, 0, 1)x(1, 1, 1, 12)12 - AIC:-423.24790042773964
ARIMA(0, 1, 0)x(0, 0, 0, 12)12 - AIC:-197.1630202950495
ARIMA(0, 1, 0)x(0, 0, 1, 12)12 - AIC:-281.26180757961856
ARIMA(0, 1, 0)x(0, 1, 0, 12)12 - AIC:-396.0827778097324
ARIMA(0, 1, 0)x(0, 1, 1, 12)12 - AIC:-421.2904222063779
ARIMA(0, 1, 0)x(1, 0, 0, 12)12 - AIC:-409.11766936393184
ARIMA(0, 1, 0)x(1, 0, 1, 12)12 - AIC:-457.5379845063193
ARIMA(0, 1, 0)x(1, 1, 0, 12)12 - AIC:-386.9831985059093
ARIMA(0, 1, 0)x(1, 1, 1, 12)12 - AIC:-415.9134512835328
ARIMA(0, 1, 1)x(0, 0, 0, 12)12 - AIC:-228.14338379997992
ARIMA(0, 1, 1)x(0, 0, 1, 12)12 - AIC:-288.6110524037759
ARIMA(0, 1, 1)x(0, 1, 0, 12)12 - AIC:-429.5754909084596
ARIMA(0, 1, 1)x(0, 1, 1, 12)12 - AIC:-440.6393618116082
ARIMA(0, 1, 1)x(1, 0, 0, 12)12 - AIC:-443.5898839668814
ARIMA(0, 1, 1)x(1, 0, 1, 12)12 - AIC:-479.98944197031284
ARIMA(0, 1, 1)x(1, 1, 0, 12)12 - AIC:-410.465706231675
ARIMA(0, 1, 1)x(1, 1, 1, 12)12 - AIC:-438.2217606024366
ARIMA(1, 0, 0)x(0, 0, 0, 12)12 - AIC:-197.0730708451228
ARIMA(1, 0, 0)x(0, 0, 1, 12)12 - AIC:-212.49122729083712
ARIMA(1, 0, 0)x(0, 1, 0, 12)12 - AIC:-454.40971132251696
ARIMA(1, 0, 0)x(0, 1, 1, 12)12 - AIC:-442.0497810841081
ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:-456.41386771386664
ARIMA(1, 0, 0)x(1, 0, 1, 12)12 - AIC:-459.0325232852657
ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:-426.479401038244
```

SARIMAX Model Building

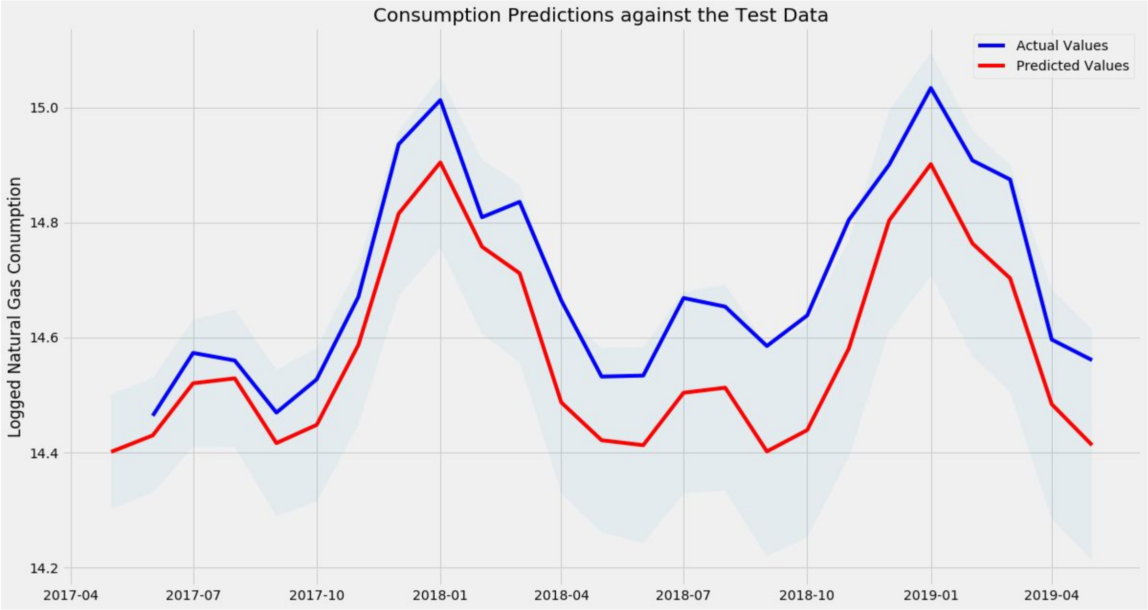
- Evaluate model results

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.5382	0.069	7.830	0.000	0.403	0.673
ma.L1	-0.9756	0.021	-45.726	0.000	-1.017	-0.934
ar.S.L12	0.9947	0.004	268.969	0.000	0.987	1.002
ma.S.L12	-0.7404	0.082	-8.995	0.000	-0.902	-0.579
sigma2	0.0022	0.000	10.575	0.000	0.002	0.003



SARIMAX Model Building

- Compare against test data
- Evaluate MSE of two datasets



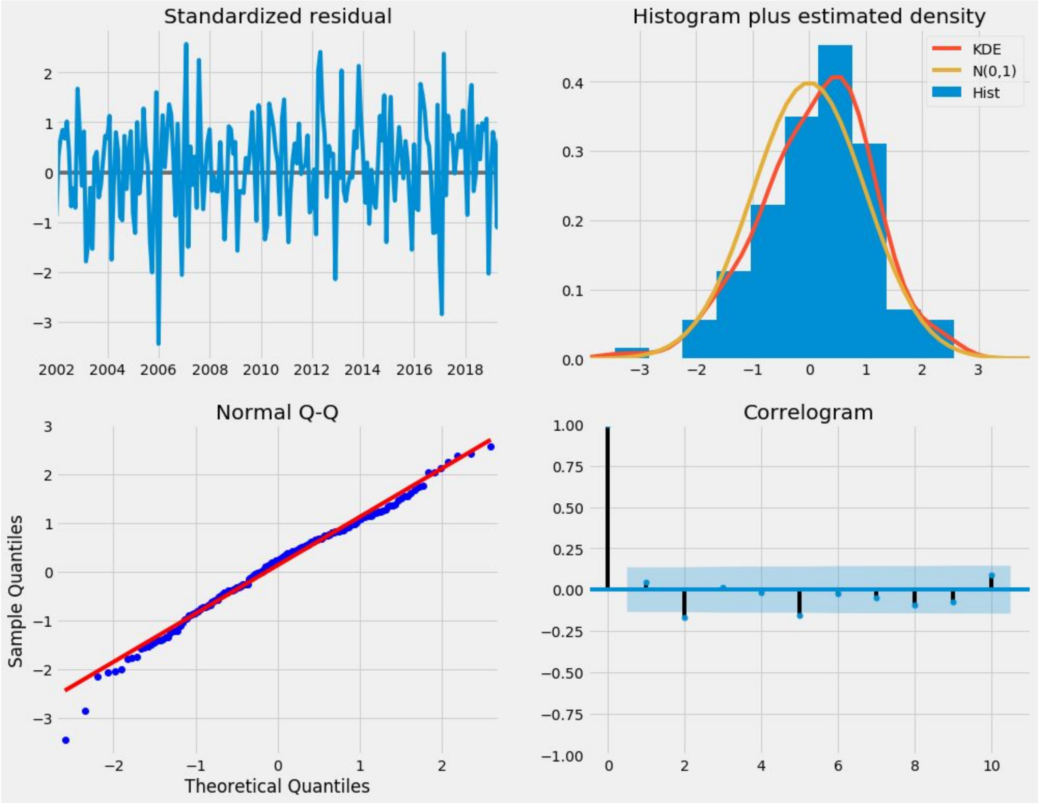
In [544]:
```python
# Compute the mean square error
def mse(predicted_means, test_data):
    mse = ((predicted_means - test_data) ** 2).mean()
    print('The Mean Squared Error of our forecasts is {}'.format(round(mse, 4)))
mse(pred_means,tes)
```

```
The Mean Squared Error of our forecasts is 0.0046
```

SARIMAX Model Building

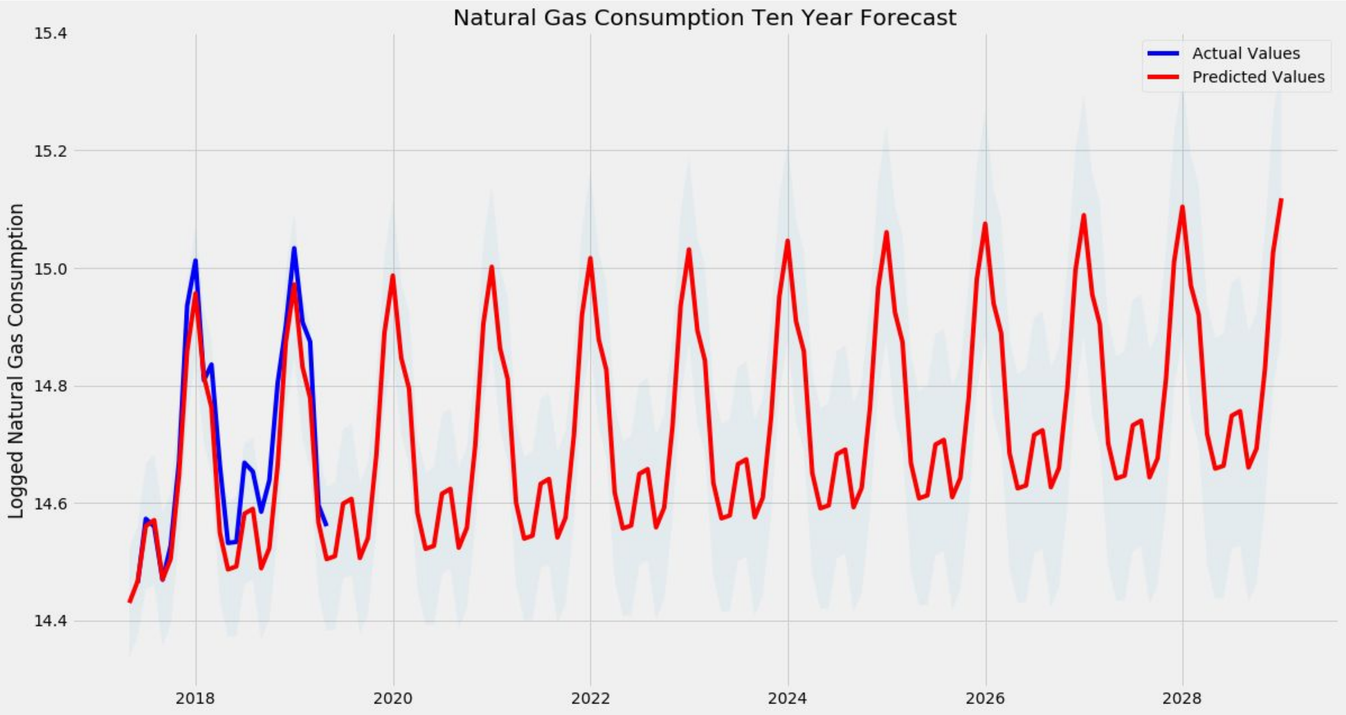
- Re-train model without hold-out groups
  - Run function to find lowest AIC
    - p,d,q = (0,1,1)
    - P,D,Q = (1,1,1)
    - M = 12
- Evaluate Results

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.4572	0.061	-7.553	0.000	-0.576	-0.339
ar.S.L12	0.9977	0.002	580.552	0.000	0.994	1.001
ma.S.L12	-0.7853	0.062	-12.654	0.000	-0.907	-0.664
sigma2	0.0024	0.000	10.699	0.000	0.002	0.003



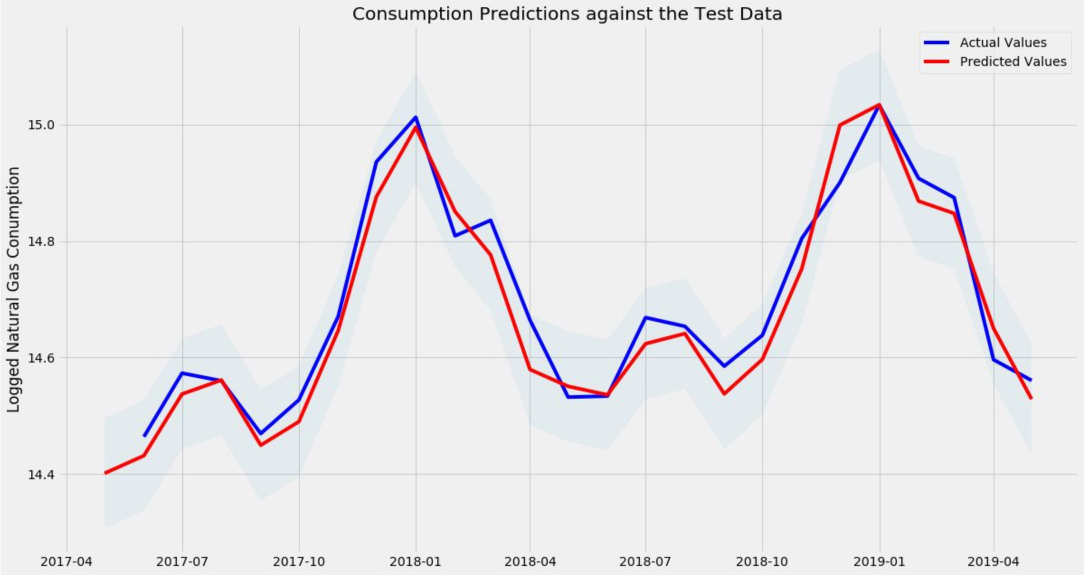
SARIMAX Model Building

- Forecast data
  - 10 year forecast



SARIMAX Model Building

- Compare against test data
- Evaluate MSE of two datasets

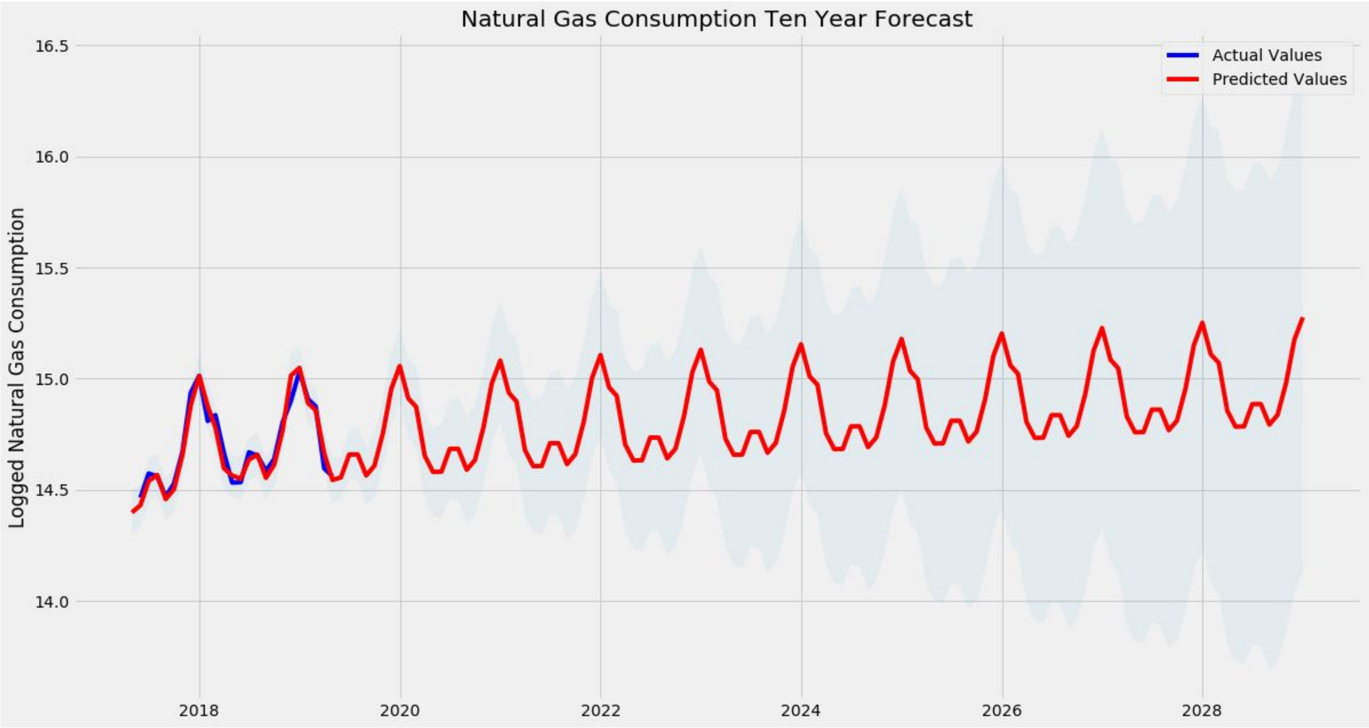



```
In [548]: # Compute the mean square error
          mse(pred_means,tes)

The Mean Squared Error of our forecasts is 0.0018
```

SARIMAX Model Building

- Forecast data
  - 10 year forecast



6. Conclusions & Future Research Opportunities

- Natural Gas Consumption is increasing over time
- Consumption is driven by price and production
- Future consumption can be predicted with reasonable accuracy using time series forecasting
- Next Steps:
  - Use API Key to evaluate other energy sources
    - Compare forecasting across various energy sources
  - Create a model that takes production and price into account for more accurate forecasting

Questions?