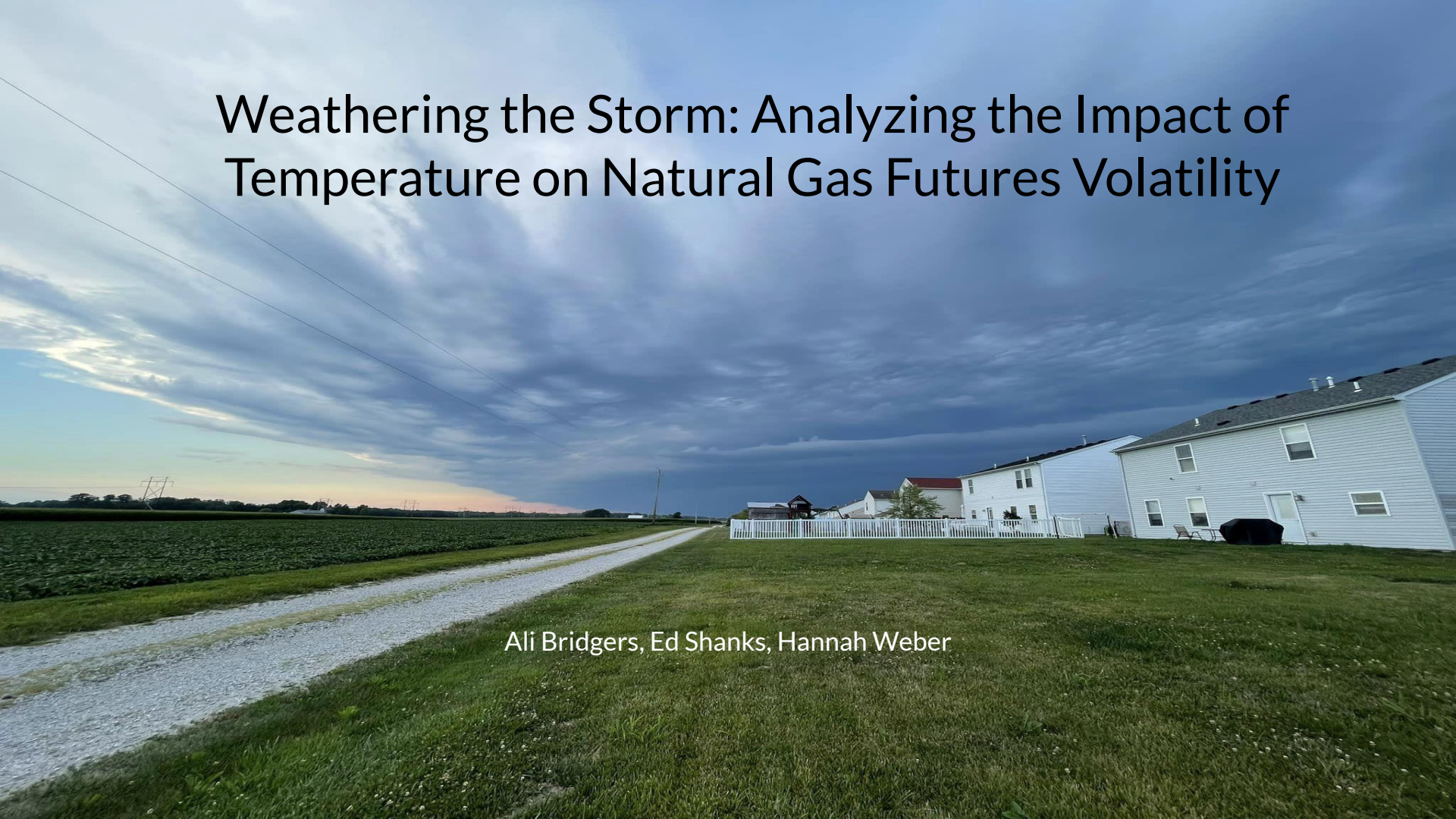
Weathering the Storm: Analyzing the Impact of Temperature on Natural Gas Futures Volatility



Ali Bridgers, Ed Shanks, Hannah Weber

How does volatility in the natural gas futures market correlate with temperature?

How did we measure volatility?

The Average True Range (ATR) Formula

The formula to calculate ATR for an investment with a previous ATR calculation is :

$$\frac{\text{Previous ATR}(n-1)+\text{TR}}{n}$$

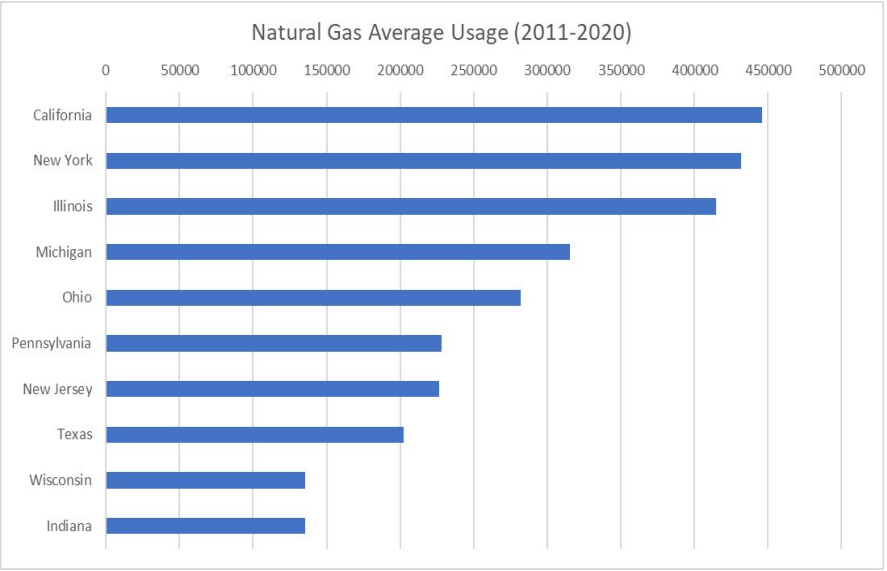
where:

n = Number of periods

TR = True range

For this project, we calculated the ATR with an exponential moving average, thereby weighing recent data more heavily. For our exponential moving average, we chose 5 day window, instead of the commonly used 14 day window, in order to better illustrate short-term volatility and account for the relatively small time window of 3 months in our datasets.

How did we determine our cities?



There was no available natural gas average usage by city, so we determined which states had the highest average usage and selected the largest city in each state (Los Angeles, New York City, Chicago, Detroit, Columbus, Philadelphia, Newark, Houston, Milwaukee and Indianapolis).

Data Collection & Sources

Natural Gas Futures and ATR

Henry Hub Natural Gas Futures (NG=F).

(https://pypi.org/project/yfinance/)
(https://finance.yahoo.com/)


```
_id: ObjectId('64b162ea539c722a3086f290')
Date: "2011-12-01"
Open: 3.575000047683716
High: 3.688999891281128
Low: 3.5409998893737793
Close: 3.6480000019073486
Adj_Close: 3.6480000019073486
Volume: 175649
ATR: 0.14800000190734863
```


Daily Weather Data

(https://rapidapi.com/meteostat/api/meteostat/)


```
_id: ObjectId('64b162e8539c722a3086cfb3')
date: "2011-12-01"
tavg: 33.6
tmin: 28.2
tmax: 39.6
city: "Los Angeles"
```

Database


```python
# Import weather data from multiple files for each city
cities = ['Los Angeles', 'New York City', 'Chicago',
          'Detroit', 'Columbus', 'Philadelphia',
          'Newark', 'Houston', 'Indianapolis', 'Milwaukee']

for city in cities:
    collection = db['weather_data']  # Use the same collection for all cities
    city_file_path = os.path.join(weather_directory, f'{city}_data.json')
    import_weather_data(city_file_path, weather_schema, collection, city)
```

```python
# Import futures data from multiple files
futures_files = os.listdir(futures_directory)
for index, file in enumerate(futures_files):
    if file.endswith('.json'):
        label = f'Winter {index + 1}'
        file_path = os.path.join(futures_directory, file)
        import_futures_data(file_path, futures_schema, futures_collection, label)
```

Web API


```python
@app.route("/<int:year>/weather_data/<city>/")
def weather(city, year):
    # Convert the city name to lowercase
    city = city.lower()

    # Determine the start and end dates based on the provided year
    start_date = f"{year}-12-01"
    end_date = f"{year+1}-02-28"

    # Query the weather data within the specified winter season
    weather_data = db.weather_data.find({
        "city": {"$regex": f"^{city}$", "$options": "i"},
        "date": {"$gte": start_date, "$lte": end_date}
    }).sort("date", 1)

    # Convert the queried data into a list of dictionaries
    weather_list = []
    for data in weather_data:
        # Get the ObjectId and convert it to a string
        data['_id'] = str(data['_id'])

        # Append the modified data entry to the list
        weather_list.append(data)

    # Return the weather data as JSON
    return jsonify(weather_list)
```

```python
@app.route("/<int:year>/futures_data")
def futures(year):
    # Determine the start and end dates based on the provided year
    start_date = f"{year}-12-01"
    end_date = f"{year+1}-02-28"

    # Query the futures data within the specified winter season
    futures_data = db.futures_data.find({
        "Date": {"$gte": start_date, "$lte": end_date}
    }).sort("Date", 1)

    # Convert the queried data into a list of dictionaries
    futures_list = []
    for data in futures_data:
        futures_list.append({
            'Year': year,
            'Date': data['Date'],
            'Open': data['Open'],
            'High': data['High'],
            'Low': data['Low'],
            'Close': data['Close'],
            'Adj_Close': data['Adj_Close'],
            'Volume': data['Volume'],
            'ATR': data['ATR']
        })

    return jsonify(futures_list)
```

Visualizations


```
// candlestick chart
Highcharts.chart('container', {
    plotOptions: {
        series: {
            // general options for all series
        },
        candlestick: {
            // shared options for all candlestick series
        }
    },
    series: [{
        // specific options for this series instance
        type: 'candlestick'
    }]
});

// ATR basic line chart
Highcharts.chart('container', {
    chart: {
        type: 'arearange',
        zoomType: 'x',
        scrollablePlotArea: {
            minWidth: 600,
            scrollPositionX: 1
        }
    },

// temperature area range chart
Highcharts.chart('container', {
    plotOptions: {
        series: {
            // general options for all series
        },
        arearange: {
            // shared options for all arearange series
        }
    },
    series: [{
        // specific options for this series instance
        type: 'arearange'
    }]
});
```

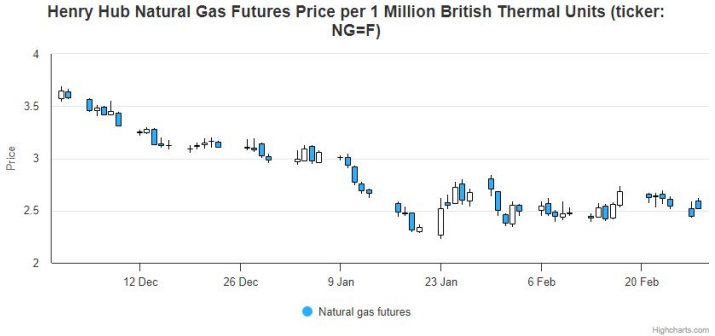
Website Demo

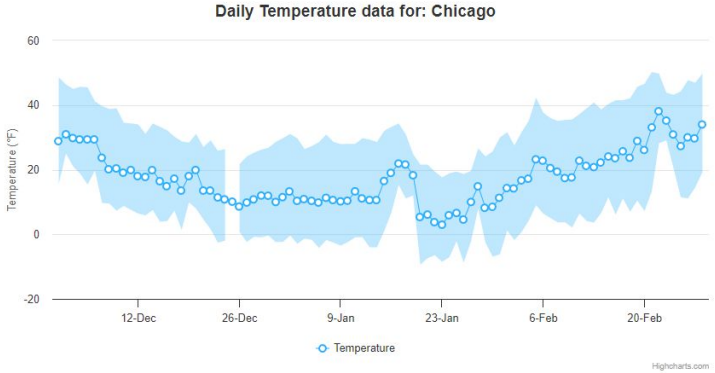
Henry Hub Natural Gas Prices and Recorded Temperatures

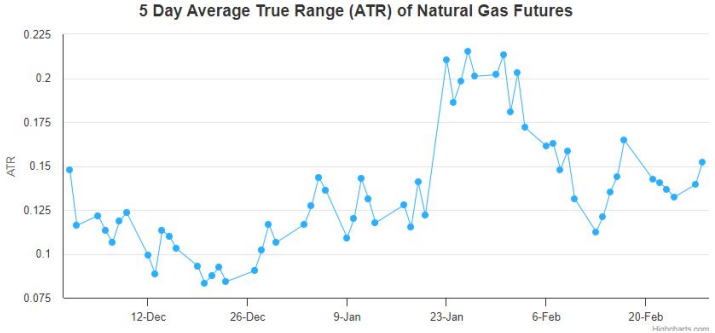
by Ali Bridgers, Ed Shanks and Hannah Weber

Change year: 2011

Change city: Chicago







Conclusion

- There is no discernible correlation between the average temperature of the selected cities and the volatility of natural gas prices.
- Although there are a few instances where volatility spikes with extremely low temperature
    - Specifically in New York City around January 21st, 2018
    - Here we can see a significant volatility spike corresponding with extremely low temperatures
    - Oddly enough, volatility continues to decrease around the time of the next cold snap around January 31st

Resources

- Meteostat API - https://rapidapi.com/meteostat/api/meteostat/
- Yahoo Finance API - https://pypi.org/project/yfinance/
- US Energy Information Administration Natural Gas Consumption by End User - https://www.eia.gov/dnav/ng/ng_cons_sum_a_EPG0_vgt_mmcf_a.htm
- HighCharts Demo Area Range and Lines - https://www.highcharts.com/demo/highcharts/arearange-line
- HighCharts Demo Basic Line - https://www.highcharts.com/demo/highcharts/line-basic
- HighCharts Demo Candlestick - https://www.highcharts.com/demo/stock/candlestick