For office use only
T1 ________________
T2 ________________
T3 ________________
T4 ________________

For office use only
F1 ________________
F2 ________________
F3 ________________
F4 ________________

2012

15th Annual High School Mathematical Contest in Modeling (HiMCM) Summary Sheet
(Please attach a copy of this page to each copy of your Solution Paper.)

Team Control Number: 3873
Problem Chosen: B

Please type a summary of your results on this page. Please remember not to include the name of your school, advisor, or team members on this page.

We propose a model that can accurately predict gasoline prices and suggest buying patterns for consumers to minimize their expenses. Starting with a multivariate regression model, we expressed gas prices in terms of economic and social data. We then incorporated the predicted gas prices in a decision-making model that informs a consumer of when to buy gas and how much to buy.
The first model predicts upcoming gas prices by looking at previous crude oil and gasoline prices, and isolating a causal relationship. We verified this relationship between multiple time series using the Granger Causality test. We were then able to extrapolate this relationship, through the use of a multivariate regression model, and combine it with Organization of the Petroleum Exporting Countries (OPEC) basket prices and Google keyword popularity data in order to create a statistically valid predictor for gas prices.

The second model uses the gas price predictions from the the regression model to suggest an optimal purchasing timeline for a consumer. In particular, this model determines the lowest-cost purchasing plan based on simple pattern recognition of gas price trends. The model follows a general rule: in order to minimize total cost, the consumer should buy more when the gas price is about to rise and avoid buying when the price is about to fall. By evaluating the prices for an extended period of time while following these guidelines, the model returns a purchase plan that is similar or identical to the lowest cost plan.

Both models alone provided us with valuable information; combining the two yields a comprehensive and accurate means of making meaningful decisions that will help consumers save money on gas in the long run. Our models were versatile enough to be applicable to a number of cities across the nation, including San Francisco, Houston, and Chicago. They provided close-to-optimal purchasing plans that are able to save the typical consumer up to $10 every year on gas.

A Predictive, Multivariate Regressive Analysis of Gas Prices and Decision-Making

November 17, 2012

Contents

1 Problem Restatement

Model gas prices using data from the weather, economy, and world events. With the model, formulate buying patterns that minimize cost for a vehicle with a sixteen gallon tank that gets an average of twenty-five miles per gallon, or four hundred miles per tank. Specifically, apply this cost analysis to a customer that purchases gasoline once a week, either filling a full tank, half tank, or none at all. Consider two scenarios: the driver travels either one hundred miles each week or two hundred miles each week.

Train the models using 2011 data from both San Francisco and Houston, and test them for accuracy against the 2012 data. Include a non-technical letter for the San Francisco Chronicle.

2 Assumptions and Justifications

1. **Every week, one can only buy a tank's worth of gas, half a tank of gas, or no gas at all.**
   Given, assumed to be exact for convenience.

2. **Consumers drive either exactly 100 or exactly 200 miles per week.**
   Given, assumed to be exact for convenience.

3. **Cars hold exactly 16 gallons of gas and get 25 miles per gallon of gas.**
   Given, assumed to be exact for convenience.

4. **Consumers can end up at a gas station with a completely empty tank.**
   Since the consumers can buy a full tanks worth of gas, they must be able to enter the gas station with no gas in their tanks.

5. **Consumers go to gas station on the same day of the week, every week.**
   This simplifies the time series in the regression model, giving it a constant time interval of a week.

6. **Relationships between gas prices and oil prices stay consistent throughout 2011 and 2012.**
   Granger causality may not imply true causality. Two time series can be driven by a third and still pass the Granger test when, in reality, one

does not cause the other. As a result, they could lose their correlation over time. However, we assume such a scenario is unlikely.

7. **No huge market disruptions occur in 2012.**
   A huge market crash has an unpredictable effect on the price of gas, and we could not predict the price if such a disaster were to occur.

8. **Gas prices, crude oil prices, and OPEC basket prices do not fluctuate very much within a one week span.**
   Though this may not be completely true, we are only given weekly data on gas prices. Also, our calculations show that gas prices tend to pivot in five week periods, suggesting that there will be little change within one week. We assume the same for crude oil prices and OPEC basket prices for the sake of simplicity.

9. **Weather has no significant effect on gas prices.**
   Research shows that weather has a less than 1% effect on gas prices.

10. **Seasons should not be specifically calculated for.**
    The effect of changing seasons should be already reflected in OPEC basket prices and crude oil prices.

11. **It is insignificant that OPEC basket prices and Google correlations do not occur on the same day as the oil prices we associate them with.**
    In the linear regression model, we can associate OPEC and Google data that is a day or two apart from oil prices without any significant long-term effects on the model. This assumption simplifies the data mining process.

12. **Inflation is negligible.**
    It does not change quickly enough to have a noticeable effect over the course of 2011 and 2012.

13. **The patterns that Google Correlate picks up in gas price data are not statistical flukes.**
    The correlations we found all have the word "mpg" in them, which suggests the correlations are not a fluke and that Google searches with "mpg" in them are accurate indicators of future gas price changes.

14. **The actions of the consumer (saving money on gas) do not have an effect on the gas price.**
    Compared to the population as a whole, small savings on gas by an individual do not have an effect on the price as a whole.

3 The Model

3.1 Model Approach

We decided it would be best to develop two separate models: one to predict the gas prices over time and one to model the consumer's decision-making process.

The consumer model takes the gas model as an input and calculates the final result. This allows us to maintain a simple and efficient solution.

3.2 Predicting Pump Prices

A critical part of our model is our ability to accurately calculate the gas prices from 1-2 weeks in the future. We do this by extrapolating current trends in crude oil, as well as other possible influences on gas prices.

For the bulk of our model, we focus on the predictive relationship between crude oil and gasoline prices. However, in order to definitively calculate gas prices, we must do three things:

1. We must prove that there is a definitive connection (causality) between our two linked variables, crude oil and gas prices

2. We must map a simple regression model, and use a statistical analysis to predict how well it fits the 2011 data, as well as how effective it is at predicting the 2012 data.

3. Finally, we must develop a better model to fit the 2011 data, as well as predict the 2012 data. To do this, we establish a heuristic linear regression model, that takes in the current gas prices on an iterative basis, and update the model to account for current trends.

3.2.1 Causality

As far as causality goes, the key relationship that we see is the one between crude oil and gas prices. In the current system, crude oil is measured in dollars per barrel, and must be processed before released as gasoline. Therefore, there is a small period of time in which we know the price of crude oil, but do not know that of gasoline. This small period of time is a 12-14 day interval, allowing us to use the crude oil to predict gasoline prices over time.

However, before we can fit a relationship, we must prove that there is an actual relationship. This means that we need to intuitively prove that Crude Oil prices actually affect Gasoline prices. That is to say, there is a definite Causality between crude oil and gasoline.

To prove Causality, we use the Granger Causality, a statistical test which tests the null hypothesis that, given x and y, that x does not cause y.

The null hypothesis is a statistical tool that basically indirectly tests a hypothesis, by disproving all alternatives to the hypothesis. Generally, the null hypothesis tests whether a statistical model is valid or not. It does this by returning a p-value, a value that discerns the probability that the model is accurate, i.e. rejects the null hypothesis.

For us, as is standard, we chose a cutoff p-value of 0.1. That means, that if our data shows a p-value less than 0.1, or is more than 90% accurate, it is a valid model.

In addition, the Granger Causality includes a secondary test, the Wald Test. The Wald test, unlike the null hypothesis, actually tests the direct causality of the two inputs (crude oil and gasoline). If the Wald Test returns a value of 1, or 0, then there is no causality (crude oil does not affect gasoline prices). However, if it returns any other value, then there is a causality and our model is valid.

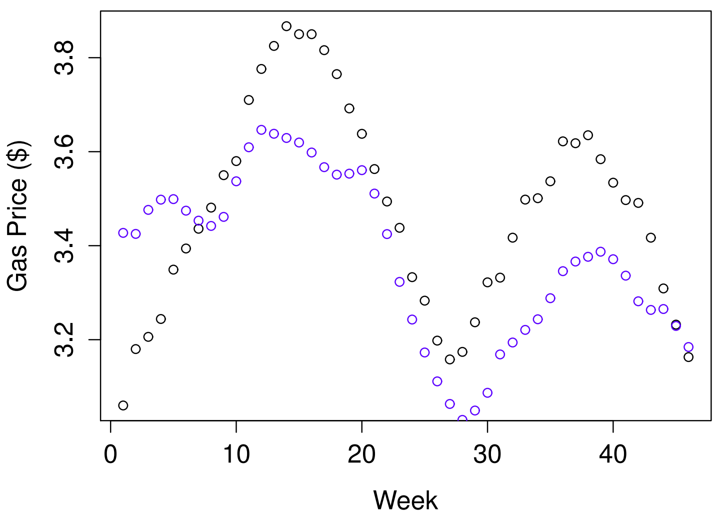

Figure 1: Initial model, with single feature–Pearson correlation 0.795

3.2.2 Basic Regression Model

Our initial attempts at constructing a model involved performing a linear regression on the crude oil prices versus gas prices 1-2 weeks later. Later attempts, in an attempt to remove the randomness found in the varying crude oil prices, we used a three-period moving average of the crude oil prices in the linear regression. Afterwards we incorporated other sources of data, such as OPEC basket prices into our linear regression. Google recently published a study about how search trends can foreshadow flu trends, and we applied that thinking to gas price trend and found a significant correlation with the keywords “better mpg”.

For each dataset, we generated several features (e.g. the price one week ago, the price two weeks ago, the moving average of the price two weeks ago, etc.) and incorporated all of them into the linear regression, all trained on 2011 data. In essence, the predicted gas price in the future can be expressed as a weighted sum of the features of the crude oil prices, etc. currently being encountered and/or seen in the past:

$$G(t+1) = \sum_{i \in F} w_i f_i(t) \tag{1}$$

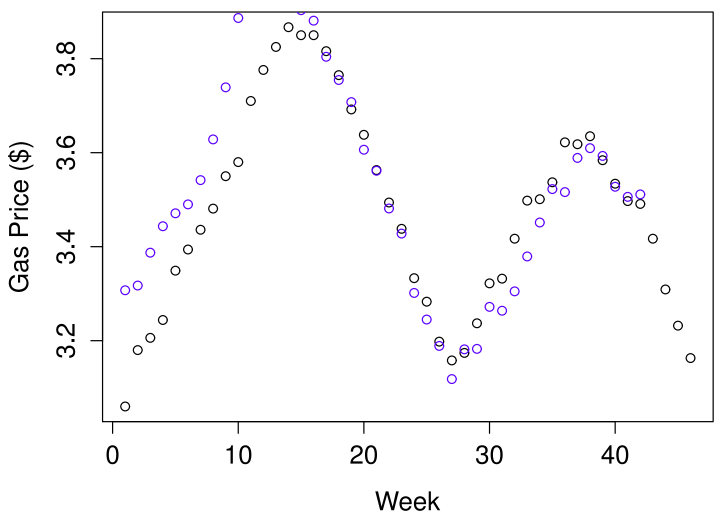

Figure 2: Model with multiple features–Pearson corrleation 0.899

3.2.3 Adaptive Regression Model

Even after incorporating several sources of data and numerous features, our predictions followed the general trend of the true gas prices (had similar shapes) but did not line up with them properly, and so was not directly comparable with gas prices. Therefore, to account for this steady-state error we considered only the change in the predicted gas prices from week to week, rather than the prediction's actual magnitude, and assumed this lined up with the change in the true gas prices from week to week–and so we calculated the future gas price by adding the predicted change in gas price to the current, true gas price.

$$G(t+1) = G(t) + f(t+1) - f(t) \tag{2}$$

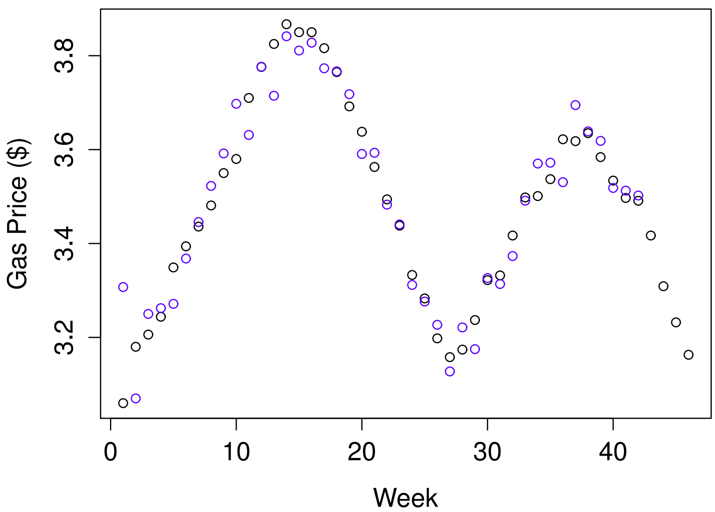

Figure 3: Model with multiple features + adaptation–Pearson 0.958

3.3 Formulating a Consumer Cost Model

3.3.1 Purchasing Trends with Finite Foresight

Given the predicted gas prices for a finite number of weeks into the future, there are simple buying principles which can be followed to minimize the consumer cost. In order to unveil these trends, we analyzed multiple situations with varying scopes of gas price foresights.

We define n as the number of weeks into the future that gas prices can be predicted.

Applying this definition of n, we get *finite foresight*, a prediction of gas prices n weeks into the future, where n must be a reasonably small number.

Regardless of n, our common principle is as follows: In order to minimize average cost per gallon, the consumer should buy more gas when the cost is about to rise and avoid buying when the cost is about to fall.

We sought to minimize an average cost per gallon, as opposed to the total cost, in a given time period because the number of gallons purchased may vary. In a 12-week period, for example, one consumer might completely deplete their gas while the other might have some remaining in her tank at the end of the 12th week. The latter consumer will have likely spent more money on gas, but will have purchased more gallons as well. An average cost per gallon normalizes for these situations.

For $n = 1$: In the simplest scenario—in which the consumer has access to predictions one week into the future—we see that we can minimize cost with the following reasoning, based on the forecasted price and current states of the tank:

Forecasted Price	Empty Tank	Half Full	Full Tank
$\uparrow$	Buy Full	Buy Half	Pass
$\downarrow$	Buy Half	Pass	Pass

Table 1: Strategy for buying gas if one week is known in advance.

This decision matrix works for both consumers driving 100 and 200 miles per week. In both cases, an empty tank forces the consumer to buy some gas—a full tank if the price is expected to increase and a half-tank otherwise. With half a tank of gas, the consumer buys as much gas as she can when prices are increasing and holds off otherwise. With a full tank, regardless of the price prediction, a consumer is unable to buy gas that week.

Special consideration is given to the consumer who drives 100 miles per week. Because her car consumes a quarter tank each week, she has two other possibilities in addition to empty/half/full tank: quarter tank and three-quarters tank. In both cases, she has no choice but to act conservatively and follow the rules for half tank and full tank respectively.

For $n = 2$: In the next scenario in which the consumer can see two weeks into the future, she is faced with several more options.

To analyze the situation, we plotted the two predicted gas prices alongside the current price. Looking at the graphs and applying our core principles, we came up with two courses of action: *reach* and *stock*. *Reach* implies purchasing only enough gas in order to *reach* the week with the lowest price, at which point the consumer should buy as much gas as possible. *Stock* implies buying gas while the costs are still low, which allows consumers to avoid buying gas on high-price days.

When a consumers tank is low, she is forced to purchase some amount of gas. In a *reach* scenario, the consumer should buy just enough gas to *reach* the lower cost. Therefore, if the gas is likely to drop one week in advance, the consumer should only buy a half tank, so as to be able to buy cheaper gas the next week. If the gas is likely to drop to its lowest two weeks in advance, then the consumer should purchase just enough fuel in order to *reach* the cheapest day (half tank for 100 mi/week and full tank for 200 mi/week).

In a *stock* scenario, the consumer should simply buy as much fuel as possible in the current week.

When a consumer's tank is half-full, she should follows the same trends as when her tank is empty. However, the residual gas constrains the consumer to purchasing a maximum of a half tank. On the other hand, the residual gas also gives the consumer the option of not buying gas at all. The result of this new condition is a shift from buying a full tank or half tank to buying a half tank or passing. (The caveat to this scenario, just like in the low tank scenario, is when the price hits its minimum on the second predicted day for the 200 miles per week system.)

When the consumer's tank is full, she is unable to fill her tank any further and is forced to pass regardless of the price.

Forecasted Price	Empty Tank	Half Full	Full Tank
$\uparrow, \uparrow$	Buy Full	Buy Half	Pass
$\uparrow, \downarrow$ (Lower)	Buy Half*	Pass*	Pass
$\uparrow, \downarrow$ (Higher)	Buy Full	Buy Half	Pass
$\downarrow, \uparrow$ (Higher)	Buy Half	Pass	Pass
$\downarrow, \uparrow$ (Lower)	Buy Half	Pass	Pass
$\downarrow, \downarrow$	Buy Half	Pass	Pass

Table 2: Strategy for buying gas if two days are known in advance. *Lower* means the 3rd price is lower than the first. *Higher* means the 3rd price is higher than the first.

*Note: For 200 miles/week, buy full and half respectively

For $n \geq 3$: For a scenario with foresight for an extended period of time—three weeks or more—we wrote a computer program that could calculate the most effective purchasing plan. The program analyzed all the possible plans and their associated cost.

We call this model, which is aware of all future gas prices, *absolute clarity.* It finds the optimal purchasing plan.

Although the program has an exponential order of growth, it runs reasonably quickly for the values of n we are interested in. We mainly worked with small integer n values because of the limitations of any model that predicts gas prices into the future. That is, n must be a small integer because most data extrapolations cannot see more than a few weeks into the future with great accuracy.

This program also helped serve as a reference tool for evaluating our models. Unlike a finite foresight, it is guaranteed to find the optimal purchasing plan.

3.3.2 Finite Foresight vs. Absolute Clarity

We developed various scenarios in which we compared both $n = 1$ and $n = 2$ finite foresight with absolute clarity, which we calculated with our program. The finite foresight models repeatedly performed closely to the optimal model. Since gasoline price trends have relatively few pivot points—since 2003, the gas price has changed direction only once in roughly every 5 weeks—finite foresight is surprisingly accurate. It is only when multiple pivots occur in a few weeks that the model fails from lack of foresight.

3.4 Application of Model to Cities

Using the linear regression, the gas prices of each of the cities Chicago, Houston, and San Francisco were expressed in terms of previously known crude oil prices, OPEC Basket prices, and the number of search hits for "better mpg" on Google.

Each of the prices as a function of time can be expressed as such:

$$G(t+1) = a_0 + a_1 O(t) + a_2 * SMA_3(O,t) + a_3 * SMA_3(C,t) + a_4 B(t-1) + a_5 * SMA_3(B,t) \quad (3)$$

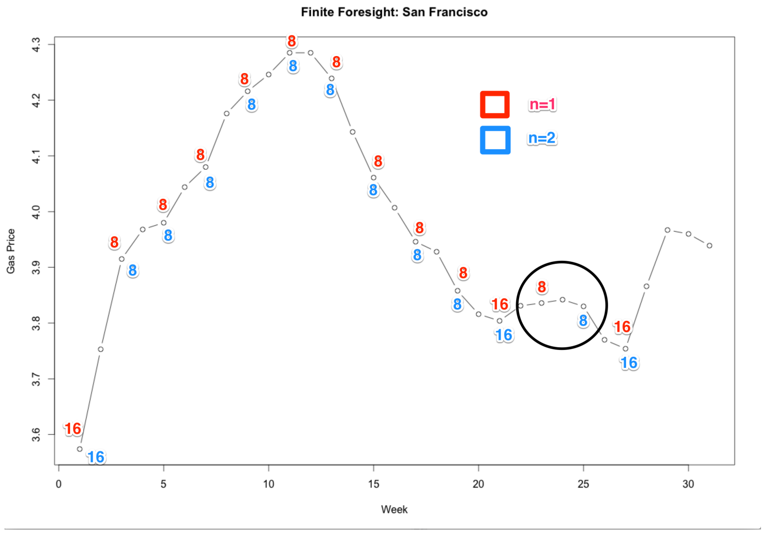

Figure 4: Comparison of Finite Foresight $n = 1$ (red) to Finite Foresight n =2 (blue). As it turns out, $n = 2$ is the exact same as an Absolute Clarity model, and $n = 1$ is only set apart by one purchase.

Where a_{0-5} are constants of regression, t represents the current week, O represents the OPEC Basket price (in USD), B represents the normalized search volume for "better mpg", C represents the crude oil barrel price (in USD), and SMA_3 represents a 3-period moving average for a function.

City	a_0	a_1	a_2	a_3	a_4	a_5
Chicago	-0.3497	0.0100	0.0135	0.0068	-0.0629	0.2508
Houston	0.6917	0.0086	0.0109	0.0035	-0.0095	0.2282
San Francisco	2.07823	0.0046	0.0161	-0.0086	-0.0148	0.3399

Table 3: City Gas Price Coefficient Table

After the model had trained on 2011 data, it was used to predict the gas prices during the first nine months of 2012. The data was then entered into the Absolute Clarity and Finite Foresight programs (see Appendix B.2). For further comparison, we modeled the buying patterns of the typical consumer (wait until tank is empty before completely refilling). The results of the programs can be seen in Table 4. Our model consistently outperformed the typical consumer and was almost always close to the optimal buying pattern. In the Chicago scenario, in which a consumer drives 100 miles each week, our model can save the user around \$17 a year more than the typical consumer.

Chicago (100 miles/week)	Chicago (200 miles/week)
$n = 2$: \$3.852/gal	$n = 2$: \$3.915/gal
perfect: \$3.836/gal	perfect: \$3.880/gal
control: \$3.934/gal	control: \$3.929/gal
Houston (100 miles/week)	**Houston (200 miles/week)**
$n = 2$: \$3.432/gal	$n = 2$: \$3.465/gal
perfect: \$3.428/gal	perfect: \$3.458/gal
control: \$3.480/gal	control: \$3.487/gal
San Francisco (100 miles/week)	**San Francisco (200 miles/week)**
$n = 2$: \$4.003/gal	$n = 2$: \$4.044/gal
perfect: \$4.020/gal	perfect: \$4.061/gal
control: \$4.067/gal	control: \$4.061/gal

Table 4: 2012: Average cost per gallon of gasoline for three different cities under two different distance scenarios. The three algorithms for finding the average cost: Finite Foresight ($n = 2$), Absolute Clarity (perfect), and regular buying patterns (control). Regular buying patterns assume only buying a full tank when out of fuel. Consumers driving 100 miles per week in Chicago will save around \$17.056 in 2012 if they follow the $n = 2$ model as opposed to the control model.

4 Strengths and Weaknesses

Strengths	Weaknesses
Our model is fairly adaptable. If we are provided with the data for each region, we are able to get a consistent accuracy in the predictions of gas prices.	Limited prediction range–cant predict prices more than two weeks in the future.
Our model is also extremely efficient. By predicting gas prices 2 weeks in advance, we are able to use our fairly efficient algorithm that hashes the data and returns the best option for every price combination.	Our model, however, does not have the ability to predict gas trends with any uncertainty. Essentially this means that our model will only follow the pattern, and will under no means predict uncertainty, which is why our assumptions include fairly predictive gas prices.
Our model is actually fairly comprehensive as it is. We pull our regression data from a large variety of sources we proved directly correlated to gas prices, such as OPEC, crude oil prices, and MPG search efficiency, allowing us to develop a predictive model.	We use a brute force approach to our decision model, meaning that with a larger set of data, we lose a huge portion of our model's efficiency. Therefore, we can only keep a maximum of two weeks stored in the model at any given time.

Table 5: Model Strengths & Weaknesses

5 Extensions

1. We would have liked to develop a more conclusive model, one that would actually take more data as inputs, for example, weather, stock market, and political data, and tracked their effects on gas prices.

2. We definitely would have liked to see how the model would have changed by changing the refill parameters, from either half a tank or a full tank, to variable tank fills. For example, you can choose to fill up your tank a quarter of the way one day, and a third of a tank another.

3. We would have liked to model instantaneous gas prices as per the opening and closing of the stock exchange. The price of gas is directly related

to the stock market, and by modeling instantaneous change, we would be able to better predict gas prices.

6 Letter to San Francisco Chronicle

November 19, 2012
To the San Francisco Chronicle:
Model Developed to Help Consumers Save Money on Gas

By this time next week, gasoline prices in San Francisco will have decreased by 1%. This prediction is just one of many findings unveiled in a recent paper of ours. In the paper, we showed that gas prices in the United States can accurately be predicted using social and economic data.

In particular, we ran an analysis on San Franciscos gas prices and discovered a simple model that consumers can follow to save considerable amounts of money. To minimize costs, consumers should stock up on gas before an increase in price and buy sparingly when a decrease is expected. Although these trends appear simple, they become increasingly more complex as one looks at prices further into the future. For this purpose, we developed a market analysis algorithm which optimizes this process for you using basic pattern recognition.

The calculations drawn from our prediction and algorithm suggest that San Franciscans can save up to \$0.26 every week. Though this might seem like an insignificant amount, the small sum quickly accrues to \$13.31 within a year.

Behind the scenes, our model utilizes publicly available data, including crude oil prices and the Organization of the Petroleum Exporting Countries (OPEC) Basket Prices. In an innovative approach, we analyzed search frequency data from Google and found certain keywords, such as better mpg, to be strong indicators of future gas prices.

As prices in the Bay Area have been decreasing recently, it is not surprising that this trend is set to continue for at least a week more. By examining the relationship between OPEC, crude oil prices, and key search terms, we have created a usable prediction model for the price of gasoline that can help inform the consumer on buying decisions. Our flexible model can predict the prices not only in San Francisco, but also in cities such as Chicago, Houston, Denver, New York, and Los Angeles.

A Data

To train the models, we used various sources of data from late 2010 to late 2011. We collected data on the cost of barrels of oil, the value of the OPEC Basket, the cost of gas per gallon in Chicago, San Francisco, Houston; and the normalized number of Google searches for the phrase "better mpg". We then tested our model with the data from 2012.

Table 6: Data used in training and testing

Date	Oil	OPEC Basket	Chicago	SF	Houston	Hits for "better mpg"
Nov 05, 2010	84.93	84.92	3.068	3.159	2.64	-0.006
Nov 12, 2010	86.91	82.35	3.076	3.207	2.676	-0.035
Nov 19, 2010	82.23	80.14	3.014	3.203	2.66	-0.078
Nov 26, 2010	82.28	83.65	2.981	3.188	2.625	-0.16
Dec 03, 2010	86.75	87.87	3.066	3.241	2.748	0.133
Dec 10, 2010	88.5	88.21	3.054	3.295	2.788	-0.174
Dec 17, 2010	88.27	89.54	3.107	3.287	2.799	-0.148
Dec 24, 2010	89.66	90.08	3.246	3.331	2.886	0.125
Dec 31, 2010	90.97	91.28	3.261	3.322	2.876	0.592
Jan 07, 2011	89.54	92.92	3.274	3.351	2.907	0.808
Jan 14, 2011	91.02	93.8	3.24	3.377	2.918	0.592
Jan 21, 2011	89.75	91.8	3.227	3.388	2.929	0.806
Jan 28, 2011	86.11	96.39	3.245	3.381	2.91	0.539
Feb 04, 2011	89.52	96.12	3.31	3.419	2.94	1.001
Feb 11, 2011	85.51	99	3.284	3.47	2.948	1.115
Feb 18, 2011	84.13	104.01	3.326	3.574	2.985	1.468
Feb 25, 2011	95.26	108.27	3.563	3.753	3.216	1.934
Mar 04, 2011	101.05	109.55	3.678	3.915	3.382	2.433
Mar 11, 2011	103.74	106.56	3.668	3.968	3.419	1.632
Mar 18, 2011	99.79	110.23	3.714	3.98	3.42	1.254
Mar 25, 2011	104.41	109.87	3.756	4.044	3.467	1.767
Apr 01, 2011	105.08	116.73	3.933	4.08	3.58	1.543
Apr 08, 2011	109.29	117.68	4.098	4.176	3.699	2.147
Apr 15, 2011	107.75	116	4.185	4.216	3.739	2.1
Apr 22, 2011	109.11	118.96	4.252	4.246	3.773	2.093
Apr 29, 2011	112.3	118.75	4.352	4.285	3.85	1.829
May 06, 2011	105.84	111.48	4.397	4.285	3.853	1.774
May 13, 2011	99.87	106.6	4.343	4.239	3.855	1.843
May 20, 2011	97.99	107.3	4.192	4.143	3.719	1.313
May 27, 2011	99.55	111.2	4.141	4.061	3.624	1.657
Jun 03, 2011	100.92	110.66	4.254	4.007	3.565	0.785
Jun 10, 2011	100.05	113.59	4.114	3.946	3.528	0.96
Jun 17, 2011	95.87	107.82	3.992	3.928	3.498	0.784
Jun 24, 2011	92.7	103.59	3.867	3.858	3.43	1.119
Jul 01, 2011	93.7	107.12	3.938	3.816	3.367	0.998
Jul 08, 2011	97.12	111.07	3.939	3.804	3.478	0.536
Jul 15, 2011	96.72	112.68	3.984	3.831	3.57	1.161
Jul 22, 2011	98.01	113.65	3.948	3.836	3.595	1.173
Jul 29, 2011	97.83	111.85	3.969	3.842	3.582	0.911
Aug 05, 2011	90.85	101.53	3.923	3.83	3.542	0.73
Aug 12, 2011	82.86	105.42	3.845	3.77	3.464	0.954
Aug 19, 2011	85.36	105.91	3.812	3.754	3.432	0.665
Aug 26, 2011	85.06	109.48	3.92	3.866	3.411	0.943
Sep 02, 2011	88.07	108.32	3.982	3.967	3.429	1.008
Sep 09, 2011	87.91	108.42	3.918	3.96	3.404	0.655
Sep 16, 2011	88.93	108.29	3.826	3.939	3.371	0.869
Sep 23, 2011	83.65	104.53	3.708	3.892	3.26	0.922
Sep 30, 2011	81.18	98.59	3.57	3.846	3.191	0.781
Oct 07, 2011	79.43	105.61	3.517	3.838	3.166	0.973
Oct 14, 2011	85.35	107.94	3.509	3.858	3.241	0.893
Oct 21, 2011	86.82	109.47	3.468	3.865	3.238	1.008
Oct 28, 2011	92.32	106.35	3.547	3.863	3.217	1.127
Nov 04, 2011	93.24	113.79	3.504	3.85	3.185	0.895
Nov 11, 2011	96.97	112.19	3.596	3.82	3.19	0.793
Nov 18, 2011	99.32	108.34	3.514	3.762	3.108	0.692
Nov 25, 2011	96.89	109.74	3.45	3.7	3.09	0.876
Dec 02, 2011	99.91	109.49	3.401	3.643	3.074	1.064
Dec 09, 2011	100.08	107.65	3.388	3.586	3.06	0.91
Dec 16, 2011	96.06	105.05	3.302	3.537	3.026	0.92
Dec 23, 2011	97.74	107.77	3.386	3.564	3.063	0.693
Dec 30, 2011	99.81	109.4	3.49	3.617	3.06	0.792
Jan 06, 2012	102.39	112.98	3.611	3.689	3.18	0.985
Jan 13, 2012	100.43	112.24	3.663	3.675	3.206	1.301
Jan 20, 2012	99.95	111.49	3.532	3.697	3.244	1.113
Jan 27, 2012	99.35	111.21	3.494	3.731	3.349	1.438
Feb 03, 2012	97.8	114.68	3.568	3.74	3.394	1.524

Continued on next page

Table 6 – *Continued from previous page*

Date	Oil	OPEC Basket	Chicago	SF	Houston	Hits for “better mpg”
Feb 10, 2012	98.56	116.63	3.522	3.814	3.436	1.85
Feb 17, 2012	101.73	119.19	3.524	4.054	3.481	2.238
Feb 24, 2012	107.18	122.14	3.787	4.313	3.55	2.688
Mar 02, 2012	107.52	122.02	4.01	4.353	3.58	2.063
Mar 09, 2012	106.32	124.64	4.13	4.356	3.71	2.546
Mar 16, 2012	106.15	123.09	4.282	4.339	3.776	1.953
Mar 23, 2012	106.41	123.54	4.47	4.336	3.825	2.203
Mar 30, 2012	105.12	122.95	4.415	4.315	3.867	1.942
Apr 06, 2012	103.52	119.38	4.285	4.257	3.85	2.328
Apr 13, 2012	102.55	116.27	4.187	4.221	3.85	1.895
Apr 20, 2012	103.15	115.8	4.212	4.191	3.816	1.69
Apr 27, 2012	103.78	117.08	4.146	4.173	3.765	1.668
May 04, 2012	103.47	109.58	4.178	4.217	3.692	1.719
May 11, 2012	96.98	108.7	4.064	4.386	3.638	1.622
May 18, 2012	93.11	106.16	3.993	4.348	3.563	1.579
May 25, 2012	90.88	105.13	3.975	4.328	3.494	1.534
Jun 01, 2012	87.06	96.19	3.894	4.271	3.438	1.745
Jun 08, 2012	84.43	94.99	3.93	4.184	3.333	1.62
Jun 15, 2012	83.27	93.73	3.827	4.05	3.283	1.595
Jun 22, 2012	81.11	90.13	3.659	3.893	3.198	1.606
Jun 29, 2012	80.23	96.44	3.551	3.795	3.158	1.755
Jul 06, 2012	85.74	96.33	3.727	3.726	3.174	1.416
Jul 13, 2012	85.78	101.29	3.719	3.734	3.237	1.648
Jul 20, 2012	90.34	100.51	3.705	3.797	3.322	1.346
Jul 27, 2012	88.88	102.22	3.819	3.816	3.332	1.45
Aug 03, 2012	89.1	107.58	4.179	3.872	3.417	1.7
Aug 10, 2012	93.14	110.1	4.159	4.152	3.498	1.485
Aug 17, 2012	94.43	112.28	4.17	4.148	3.501	1.587
Aug 24, 2012	96.22	110.22	4.093	4.166	3.537	1.616
Aug 31, 2012	95.68	112.85	4.251	4.197	3.622	2.026
Sep 07, 2012	95.68	112.68	4.185	4.196	3.618	1.814
Sep 14, 2012	97.56	110.95	4.231	4.165	3.635	1.499
Sep 21, 2012	93.7	107.99	4.117	4.192	3.584	1.796
Sep 28, 2012	91.35	109.32	4.011	4.216	3.534	1.582
Oct 05, 2012	90.81	109.46	4.001	4.675	3.497	1.748
Oct 12, 2012	91.42	111.09	3.868	4.615	3.491	1.667
Oct 19, 2012	91.59	105.94	3.668	4.441	3.417	
Oct 26, 2012	86.35	106.12	3.581	4.196	3.309	
Nov 02, 2012	85.87	105.79	3.524	4.01	3.232	
Nov 09, 2012	86.21	105.97	3.563	3.9	3.163	

B Code

B.1 Prediction model (in the R programming language):

Our prediction models were based off the multivariable linear regression function found in R–we converted data into matrices and graphed them against each other to search for trends. After finding suitable features and functions for a particular city, we can use that to extrapolate into the future. Additionally, we performed a Granger causality test to verify the correlation between different data sets and gas prices.

```r
Houston = as.numeric(as.matrix(HoustonGasPrices)
    [464:926])
Chicago = as.numeric(as.matrix(ChicagoGasPrices)
    [464:926])
SF = as.numeric(as.matrix(SFGasPrices)[464:926])
CO = as.numeric(as.matrix(CrudeOil)[464:926])
DJI = as.numeric(as.matrix(DJIA)[464:926])
OPC = as.numeric(as.matrix(OPEC)[464:926])
```

```
#Offsets a vector by n spaces, used to offset
#the prediction from the actual model.
offset <- function(x, n){
  ret = numeric(length = 5)
  length = length(x)
  if(n == 0)
    return(x)
  if(n < 0){
    n = -n
    for(i in (n+1):length)
      ret[i] = x[i+n]
    return(ret)
  }

  if(n>0){
    for(i in 1:(length - n))
      ret[i+n] = x[i]
    return(ret)
  }
}

#Isolates the parts of the vector
#within the year 2011.
Sec2011 <- function(x){
  return(x[366:417])
}

#Isolates the parts of the vector
#within the year 2012.
Sec2012 <- function(x){
  return(x[418:463])
}



#The estimate drifts from the actual price, so we
#counteract this by using only the changes in
#predicted values to create our prediction and
#build on previously encountered gas prices.
FilterDrift <- function(predicted, actual){
```

```
  filtered = numeric(length = length(predicted))
  filtered[1] = predicted[1]
  for(i in 2:length(filtered))
    filtered[i] = actual[i-1] +
    predicted[i] - predicted[i-1]
  return(filtered)
}


### SF PREDICTION ###
SFReg = lm(formula = Sec2011(SF) ~
             Sec2011(offset(OPC, 1)) +
             Sec2011(offset(SMA(OPC,3), 1)) +
             Sec2011(offset(SMA(CO, 3), 1)))

#constants are from outputs of SFReg
predictedSF2012 = .0529299 +
  Sec2012(offset(OPC,1))*.014879 +
  Sec2012(offset(SMA(OPC,3),1))*.014710 +
  Sec2012(offset(SMA(CO,3),1)*.001502)

FilteredSF2012 =
  FilterDrift(predictedSF2012,Sec2012(SF))

### SF GOOGLE PREDICTION ###
SFGoogReg = lm(formula = Sec2011(SF) ~
             Sec2011(offset(OPC, 1)) +
             Sec2011(offset(SMA(OPC,3), 1)) +
             Sec2011(offset(SMA(CO, 3), 1)) +
             Sec2011(offset(cbettermpg, 1)) +
             Sec2011(offset(SMA(cbettermpg, 3), 1)))

predictedGoogSF2012 = 2.078237 +
  Sec2012(offset(OPC,1))*.004562 +
  Sec2012(offset(SMA(OPC,3),1))*.016097 +
  Sec2012(offset(SMA(CO,3),1))*(-.008573) +
  Sec2012(offset(cbettermpg,1))*(-.014835) +
  Sec2012(offset(SMA(cbettermpg,3),1))*.339855

FilteredGoogSF2012 =
  FilterDrift(predictedGoogSF2012,Sec2012(SF))
```

```
### HOUSTON PREDICTION ###
HoustonReg = lm(formula = Sec2011(Houston) ~
                  Sec2011(offset(OPC, 1)) +
                  Sec2011(offset(SMA(OPC,3), 1)) +
                  Sec2011(offset(SMA(CO, 3), 1)))

#constants are from outputs of reg
predictedHouston2012 = -.349702 +
  Sec2012(offset(OPC,1))*.015521 +
  Sec2012(offset(SMA(OPC,3),1))*.009919 +
  Sec2012(offset(SMA(CO,3),1)*.010307)

FilteredHouston2012 =
  FilterDrift(predictedHouston2012,Sec2012(Houston))

### Houston GOOGLE PREDICTION ###
HoustonGoogReg = lm(formula = Sec2011(Houston) ~
                  Sec2011(offset(OPC, 1)) +
                  Sec2011(offset(SMA(OPC,3), 1)) +
                  Sec2011(offset(SMA(CO, 3), 1)) +
                  Sec2011(offset(cbettermpg, 1)) +
                  Sec2011(offset(SMA(cbettermpg, 3),
                      1)))

predictedGoogHouston2012 = 0.691748 +
  Sec2012(offset(OPC,1))*.008575 +
  Sec2012(offset(SMA(OPC,3),1))*.010861 +
  Sec2012(offset(SMA(CO,3),1))*.003531 +
  Sec2012(offset(cbettermpg,1))*(-.009546) +
  Sec2012(offset(SMA(cbettermpg,3),1))*.228164

FilteredGoogHouston2012 =
  FilterDrift(predictedGoogHouston2012,Sec2012(
      Houston))

### CHICAGO PREDICTION ###
ChicagoReg = lm(formula = Sec2011(Chicago) ~
                  Sec2011(offset(OPC, 1)) +
                  Sec2011(offset(SMA(OPC,3), 1)) +
                  Sec2011(offset(SMA(CO, 3), 1)))
```

```

#constants are from outputs of reg
predictedChicago2012 = -.349702 +
  Sec2012(offset(OPC,1))*.015521 +
  Sec2012(offset(SMA(OPC,3),1))*.009919 +
  Sec2012(offset(SMA(CO,3),1)*.010307)

FilteredChicago2012 =
  FilterDrift(predictedChicago2012,Sec2012(Chicago))

### CHICAGO GOOGLE PREDICTION ###
ChicagoGoogReg = lm(formula = Sec2011(Chicago) ~
                      Sec2011(offset(OPC, 1)) +
                      Sec2011(offset(SMA(OPC,3), 1)) +
                      Sec2011(offset(SMA(CO, 3), 1)) +
                      Sec2011(offset(cbettermpg, 1)) +
                      Sec2011(offset(SMA(cbettermpg, 3), 1)))

predictedGoogChicago2012 = 0.387814 +
  Sec2012(offset(OPC,1))*.010001 +
  Sec2012(offset(SMA(OPC,3),1))*.013479 +
  Sec2012(offset(SMA(CO,3),1))*.006772 +
  Sec2012(offset(cbettermpg,1))*(-.062880) +
  Sec2012(offset(SMA(cbettermpg,3),1))*.250794

FilteredGoogChicago2012 =
  FilterDrift(predictedGoogChicago2012,Sec2012(Chicago))

grangertest(CO2011, SF2011, order = 6, na.action = na.omit)
grangertest(CO2011, Houston2011, order = 10, na.action = na.omit)
```

B.2 Customer decision heuristic (in Python):

In order to measure the effectiveness of our models, we built a program in Python. It features two functions which both calculate the optimal purchasing plans, but in different ways. In particular, they return ConsumerPlans, a custom data structure which contains a list of the purchases over time and the average price paid per gallon.

absolute_clarity() returns a list of ConsumerPlans, which can then be sorted by the average price in descending order. This allows us to determine the optimal purchasing plan. (Absolute clarity assumes accurate predictions for entire time period of calculation. This function is not realistic because of limited price prediction data, but it acts a control for us to compare our solution to.)

finite_foresight() returns a single ConsumerPlan, which is built using a $n = 2$ finite foresight model. (A finite foresight model assumes that the consumer can only predict gas prices n weeks into the future.) Of the four price values used in the $n = 2$ model each week, two are the upcoming prices from *prices_predicted* and one is the current price from *prices_actual*. The *prices_predicted* values and current price are analyzed for trends using basic pattern recognition. The *prices_actual* value is also used for the running total in the ConsumerPlan.

```python
# Fuel tank constants
FULL = 16 # gallons per tank
HALF = FULL / 2
NO_GAS = 0
# Gallons used per week (for a consumer driving 100 mi/wk)
GALLONS_PER_WEEK = 8 # 100 mi/wk * 1 gal/25 mi = 4 gal/wk

class ConsumerPlan:
    """Manage the path (purchasing plan) taken, the associated cost, and
    remaining gallons left in the tank at the end.
    """
    def __init__(self, path, averagePrice):
        # List of gas purchases (measured in gallons) over time.
        self.path = path

        # Average price of gallon during path.
```

```
        # (cost of path / gallons purchased)
        self.averagePrice = averagePrice

    def __str__(self):
        roundedPrice = str(round(self.averagePrice, 3)) # for readability
        return "Price: " + roundedPrice + "\tPath: " + str(self.path)

priced_paths = []
def absolute_clarity(prices, path, cost, tank):
    """Return an unsorted list of ConsumerPlans.

    Find the optimal purchasing plan, given the ability to see the entire
    future.

    Arguments:
        prices - a list of floats representing the cost per gallon per week
        path - a list of gas purchases in this plan so far (starts as empty)
        cost - a running total of money spent on gas purchases
        tank - amount of gas at the current week
    """
    # Base case for when the list of prices/weeks has been exhausted.
    index = len(path)
    if (index >= len(prices)):
        averagePrice = cost / (len(prices) * GALLONS_PER_WEEK + tank)
        return [ConsumerPlan(path, averagePrice)]

    consumer_plans = []
    def get_consumer_plans(amount):
        """Return list of ConsumerPlans given an amount of gas to buy for the
        current week.

        Arguments:
```

```
            amount - FULL, HALF, or NO_GAS
        """
        new_path = path[:] # duplicate path
        new_path.append(amount) # add latest purchase to path
        new_cost = cost + prices[index] * amount # accure cost

        # Add gallons purchased, while subtracted travel usage
        new_tank = tank + amount - GALLONS_PER_WEEK

        return absolute_clarity(prices, new_path, new_cost, new_tank)

    # Fill consumer_plans with all possible plans.
    if (tank + FULL <= FULL):
        consumer_plans.extend(get_consumer_plans(FULL))
    if (tank + HALF <= FULL):
        consumer_plans.extend(get_consumer_plans(HALF))
    if (tank >= GALLONS_PER_WEEK):
        consumer_plans.extend(get_consumer_plans(NO_GAS))

    return consumer_plans

def finite_foresight(prices_predicted, prices_actual, path, cost, tank):
    """Return the optimal ConsumerPlan according to the n=2 finite foresight
    model.

    The ConsumerPlan will have 2 fewer weeks than the number of prices because
    the last 2 cannot be predicted due to lack of future prices.

    Arguments:
        prices_predicted - a list of forecasted gas
```

```
            prices to be used in the
              model; revealed two weeks ahead of current week
        prices_actual - a list of actual gas prices; only revealed on the
              current week
        path - a list of gas purchases in this plan so far (starts as empty)
        cost - a running total of money spent on gas purchases
        tank - amount of gas at the current week
    """
    tank = 0
    path = []
    cost = 0
    for i in range(len(prices_predicted) - 2):
        a = prices_actual[i]
        b = prices_predicted[i+1]
        c = prices_predicted[i+2]

        amount = 0
        if tank == NO_GAS:
            if (a <= b and b <= c) or (a <= b and b >= c and a <= c):
                amount = FULL
            elif (a <= b and b >= c and c<=a and GALLONS_PER_WEEK == HALF):
                # Special case for 200 mi/wk consumer
                amount = FULL
            else:
                amount = HALF
        elif tank <= HALF:
            if (a <= b and b <= c) or (a <= b and b >= c and a <= c):
                amount = HALF
            elif (a <= b and b >= c and c<=a and GALLONS_PER_WEEK == HALF):
                # Special case for 200 mi/wk consumer
                amount = HALF
```

```
            else:
                amount = NO_GAS
        else:
            amount = NO_GAS

        path.append(amount)
        tank += amount - GALLONS_PER_WEEK
        cost += prices_actual[i] * amount

    averagePrice = cost / ((len(prices_predicted) - 2) * GALLONS_PER_WEEK +
        tank)
    return ConsumerPlan(path, averagePrice)

def print_plans(plans):
    """Print ConsumerPlans."""
    for plan in plans:
        print(plan)

def sorted_plans(plans):
    """Return ConsumerPlans sorted by price from high to low."""
    return sorted(plans, key=lambda plan: plan.averagePrice, reverse=True)

# Run algorithms
prices = [3.158, 3.177, 3.192, 3.208, 3.181] # sample data
predictions = [3.159, 3.173, 3.190, 3.200, 3]

# Generates all possible purchase plans, sorts them, and prints them in
# descending order by cost.
plans = absolute_clarity(prices, [], 0, 0)
print_plans(sorted_plans(plans))

# Prints the optimal purchase plan as determined the finite foresight model.
print(finite_foresight(predictions, prices, [], 0, 0))
```

References

[1] Balke, N. S. (n.d.). Crude oil and gasoline prices: An asymmetric relationship? Dallas Fed. Retrieved November 17, 2012, from <http://www.dallasfed.org/assets/documents/research/er/1998/er9801a.pdf>.

[2] Crude Oil 1 Year Chart. (n.d.). Crude Oil Price Chart. Retrieved from <http://gasbuddy.com/Crude_Chart.aspx>.

[3] Econbrowser. (n.d.). : Commodity Index Funds and Oil Prices. Retrieved November 17, 2012, from <http://www.econbrowser.com/archives/2012/05/commodity_index_1.html>.

[4] Granger causality. (n.d.). - Scholarpedia. Retrieved November 17, 2012, from <http://www.scholarpedia.org/article/Granger_causality >.

[5] How Gas Prices Work. (n.d.). HowStuffWorks. Retrieved November 17, 2012, from <http://auto.howstuffworks.com/fuel-efficiency/fuel-consumption/gas-price.htm>.

[6] King, K., Deng, A., Metz, D. (2012, January). An econometric analysis of oil price movements. BatesWhite. Retrieved November 17, 2012, from <http://www.bateswhite.com/media/pnc/4/media.444.pdf >.

[7] Mohebbi, M. (2011, June 9). Google correlate whitepaper. Google.com/correlate. Retrieved November 17, 2012, from <http://www.google.com/trends/correlate/whitepaper.pdf>.

[8] Statistics Lecture 8. (n.d.). Caltech. Retrieved November 17, 2012, from <http://www.hss.caltech.edu/ mshum/stats/lect8.pdf>.

[9] U.S. Energy Information Administration - EIA - Independent Statistics and Analysis. (n.d.). U.S. Energy Information Administration (EIA). Retrieved from <http://www.eia.gov/dnav/pet/pet_pri_gnd_a_epmr_pte_dpgal _w.htm >.

[10] What Affects Oil Prices? (n.d.). What Affects Oil Prices? Retrieved November 17, 2012, from <http://oilprice.com/Energy/Oil-Prices/What-Affects-Oil-Prices.html>.

[11] What Determines Gas Prices? (n.d.). What Determines Gas Prices? Retrieved from <http://www.investopedia.com/articles/economics/08/gas-prices.asp>.