Chunyang Ding

Mr. Millhollen

Gifted Precalculus

25 April 2012

Modeling Sunrises in New York

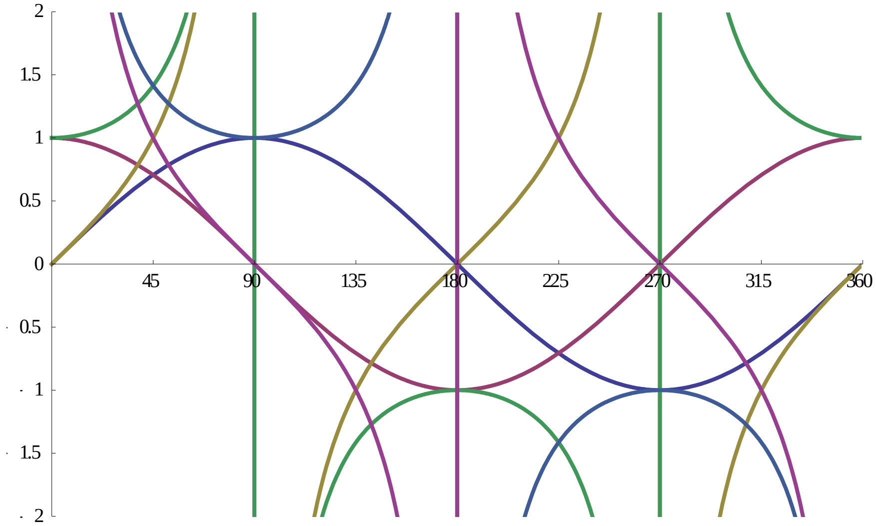
Picture this: You are a proud citizen of New York City, and suddenly, the city cries for your help. The recent recession has caused the budget to shrink, and they need to cut back at all spending. Some big shots want to shut down the entire streetlight program, on the basis that the lights are often when it's light outside, wasting precious energy and money. How could we make this process more efficient and help the world?

In order to solve this problem, the city is asking us to create a model based on some 2003 data that they had lying around. Therefore, our first task is this: how could we create a model for the sunrises of New York?

Before any work is done on actually analyzing this data, let us first consider the problem that we are facing. Right now, we are trying to find a formula that would give us the time of the sunrise given any specific day. What does this question have to do with the real world? First, we can notice that this problem is dealing with the sun. This may seem obvious, but what do we know about the movement of the sun? Through common knowledge, we know of four specific points throughout the year of significance to the sun, the winter solstice, spring equinox, summer solstice, and the fall equinox. We also know that the shortest day of the year, and also the latest day of sunrise for the year falls on the winter solstice, while the longest day of the year, as well as the earliest day of sunrise for the year, falls on the summer solstice. Given this knowledge, we turn to our data with a bounce in our step. Of course, this graph must deal with absolute values! [Insert picture here]After all, this data obviously has a high point (the winter solstice), a low point (the summer solstice), and it repeats every year! Wait, it repeats every year? Well of course, we all know that after the fall equinox comes the winter solstice again, and then the spring equinox again, and on and on. Can an absolute value function do that, as in move from a high point back to a low point and back again, iterating through many cycles? No. In that case, what do you believe would work? In other words, what *periodic* functions could we use to model the sunrise over New York City?

Let's consider the periodic functions that we know. There is the sine function, the cosine function, the tangent function, not to mention the cosecant function, the secant function, and the cotangent functions. (Fig. 1)

Fig. 1

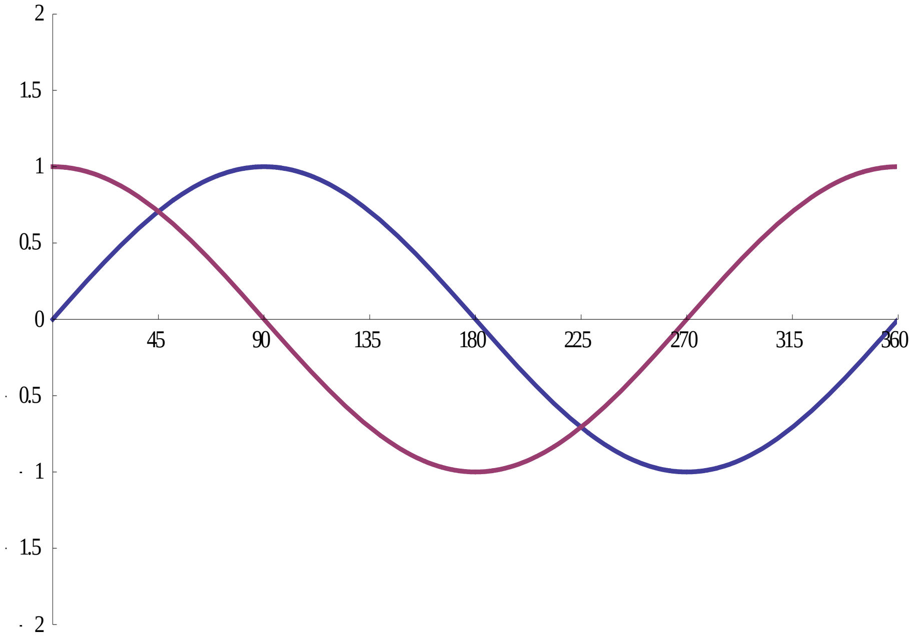


$\sin(x)$	Blue
$\cos(x)$	Violet
$\tan(x)$	Yellow
$\csc(x)$	Light Blue
$\sec(x)$	Green
$\cot(x)$	Dark Purple

That's a lot of functions, but which ones can we use? First of all, we can eliminate both the tangent and cotangent functions due to how they both have asymptotes. Consider what an asymptote would mean in this kind of situation. It means that after some period, we would have a day that has its sunrise either in infinity hours or negative infinity hours. This is clearly not possible, and therefore, we can eliminate it. We can make similar arguments for cosecant and secant, as they too have asymptotes. We are now left with two functions: the sine function and the cosine function. Which one should we use to model this?

Before we choose one of these two, let us first look at the graphs of the sine function and the cosine function. (Fig. 2)

Fig. 2



$\sin(x)$	Blue
$\cos(x)$	Violet

As we can see, the two functions are obviously related, as it seems that the cosine function is just a sine function shifted 90 degrees to the left. This means that either one of these two functions can be used to model, as if we want to change one of these functions to the other, all we have to do is apply a horizontal shift. Let us look back at the data. We see that the winter solstice, or the high point, is close to the start of the cycle. In fact, the winter solstice falls near the day of January 21st, which is very close to January 1st. Which function has its high point at the very start of the graph? That's right, the cosine function. This function begins at positive one, drops down to negative one, and then goes back up.

Congratulations, we have chosen a function to use. However, how can we apply the data to this graph? As we search through our data, we notice that there is a problem: All of the data is in hours and minutes, and we can't use hours and minutes in graphing. There is a huge problem with converting from base sixty to base ten. Therefore, what can we do in order to have data that the computer can understand? While it is possible to convert each piece of data into decimal form so that it would be $hour.fraction\ of\ hour$, this would be very taxing if you were to analyze a very large amount of data, as you would have to find each calculation by hand, or by typing in the numbers one by one. Is there an easier way to get readable data? What other format could we use that would be easy to write a formula for? One thing that we could do is to convert from military time, where if AB is the hour and CD is the minute, then the time is $ABCD$. This format is easy to calculate if we think about this logically. If we can isolate AB from this number, we would have the number of hours, which we can then multiply by sixty to get a part of the time. How could we do this? First, divide $ABCD$ by 100, and then use the floor function to get rid of anything after the decimal. Finally, multiply this by sixty, and we have the number of minutes within the hour! Algebraically, we can see this as $AB = Floor[\frac{ABCD}{100}] * 60$, assuming that $Floor[]$ is the floor function. Now, how can we isolate the minutes, or CD? We can do so by subtracting $AB00$ from $ABCD$. Thus, $AB00 = Floor\left[\frac{ABCD}{100}\right] * 100$, and we come up with $CD = ABCD - Floor\left[\frac{ABCD}{100}\right] * 100$. We can now put together the complete formula, and let's call it $a[x]$. $a[x] = Floor\left[\frac{x}{100}\right] * 60 + x - (Floor\left[\frac{x}{100}\right] * 100)$.

Using this formula, let's generate some data just to test it. Here, I will input some times throughout the day, and allow for our formula to convert them into minutes (Fig. 3).

(Fig. 3)

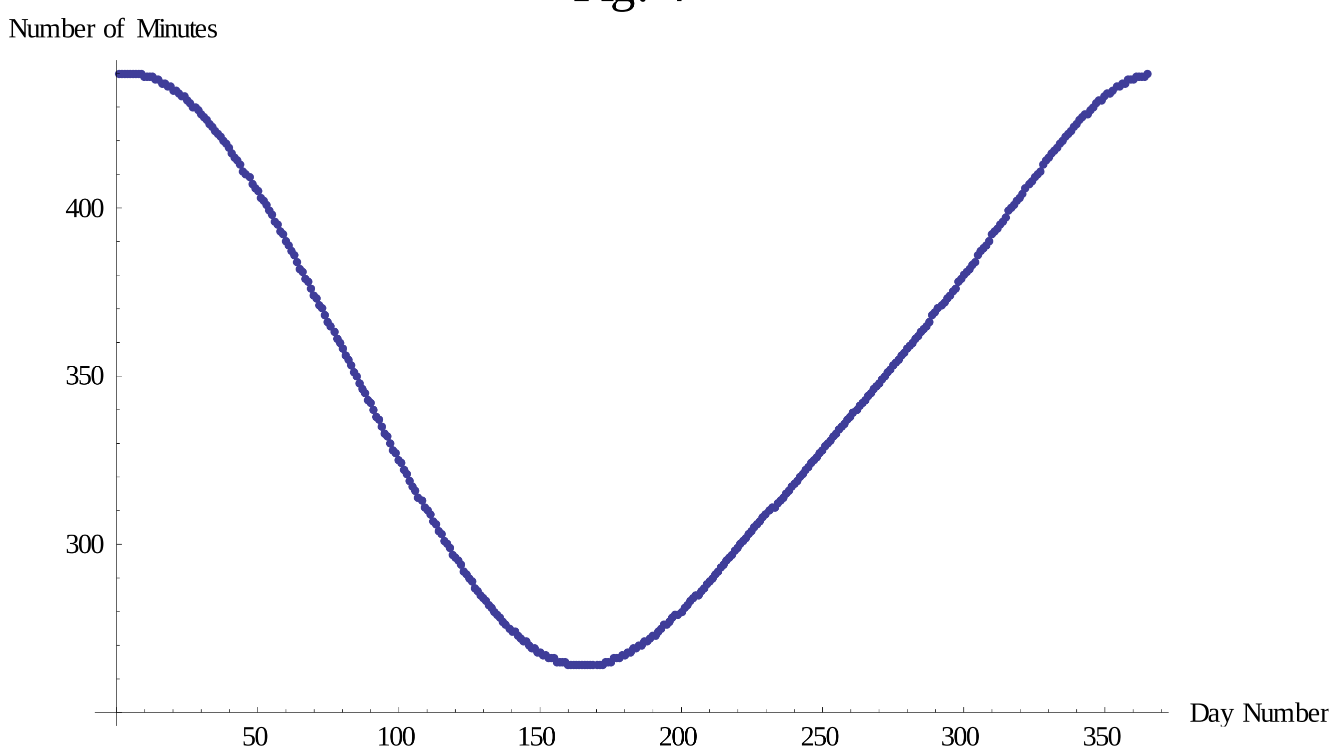
x(military time)	$f(x)$(minutes)
730	450
825	505
920	570
1015	615
1115	675
1150	710
1245	765
1340	820
1435	875

While the data provided may be enough to model a basic form of the sunrise equation, if we wish to be more exact, we must find even more points, such as finding the time of sunrise for each and every day in 2003. There is a very convenient resource that we are able to use for finding all of this data. This resource is the Naval Oceanography Portal powered by the Navy, and they keep records of the official sunrise times. [Insert picture here]The way that they store this data is easily accessible, and can be converted into data that software can read very easily. Because I will be using Mathematica 8 to graph, regress, and analyze my data, I need to import all the data into it. The way that I found the easiest was to first store all the numbers into a plain text file, import the data into Microsoft Excel using built in functions, and then converting the Excel spreadsheet into data in Mathematica. This may seem convoluted at a glance, but consider how much quicker this process is to having to input each value by hand. It really does make a difference.

After we have our data, we use our stored $a[x]$ function to modify each of the values to create a value in minutes. This becomes our new data, and we now have workable data.

Now that we have all of our data, we can get a first look at what the equation looks like. By entering the command $ListPlot[SunriseData2003NY\ ]$, we see Figure 4:

Fig. 4

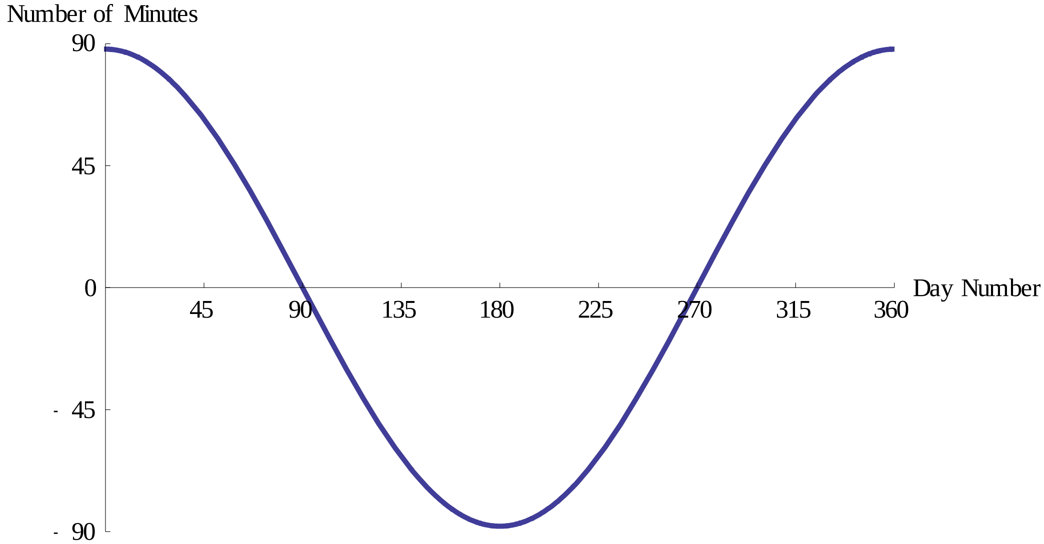


This picture definitely shows the characteristics of a cosine function, doesn't it? We see a maximum region and a low region. Note that in this, we do not actually see a definitive point of maximum or minimum time. This inaccuracy is most likely due to how the time is recorded in minutes, and at the level of specificity which we are at, the only way to see the minimum point is by observing seconds. Disregarding this, we now begin analyzing our function.

What does it mean to model a function? By modeling a function, we apply different shifts and stretches to the original function, in order to create a final function. For a simpler case such as a linear function, we have the generic formula of $f(x) = Ax + B$, where A is how slanted the line is and B is how much the graph is shifted vertically. For cosine functions, we have the generic formula of $y = A\cos\big(B(x-k)\big) + h$, where A is the vertical is stretch (modifying the amplitude), B is the horizontal stretch (modifying the period), h is the vertical transformation, and k is the horizontal transformation.

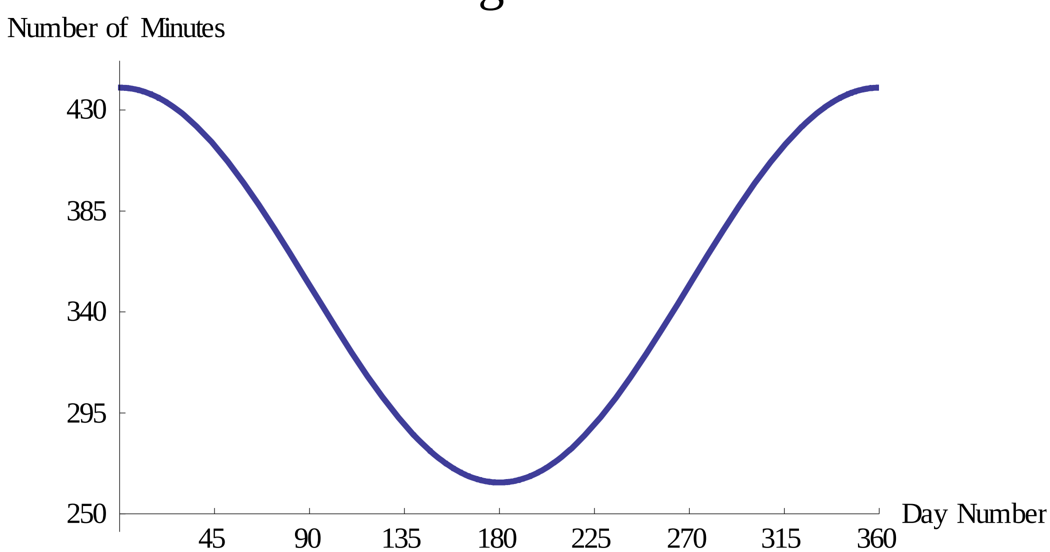
In order to correctly model this function, we must find the amplitude and the period. Of these two, the easiest is the amplitude. We can find within our data that the latest time that the sun rose was at 720, or at 440 minutes. The earliest time that the sun rose was at 424, or at 264 minutes. By subtracting the minimum time from the maximum time, we can see that it is equal to 176 minutes. However, the distance from the trough(minimum) to the crest(maximum) is not the amplitude, but is actually two times the amplitude, making our amplitude 88 minutes. Because we know that the amplitude of a sine function is one, we need to modify the function so that the amplitude is 88. The way to do this, as seen by our sample function above, is to change A to 88. See Fig. 5.

Fig. 5



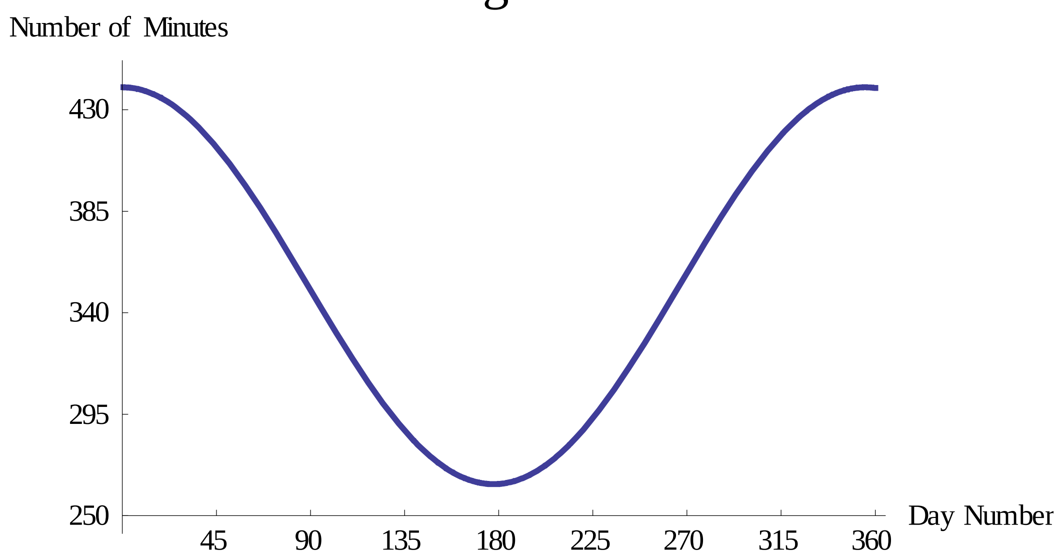
The next part of the function that we can modify is the h variable. Because we know that we must move the crest of the modified function to reach 440, and that the crest of the current function is at 88, we must move the entire function up by 352 minutes. This vertical shift translates into a change in the h value, making $h = 352$. See Fig. 6.

Fig. 6



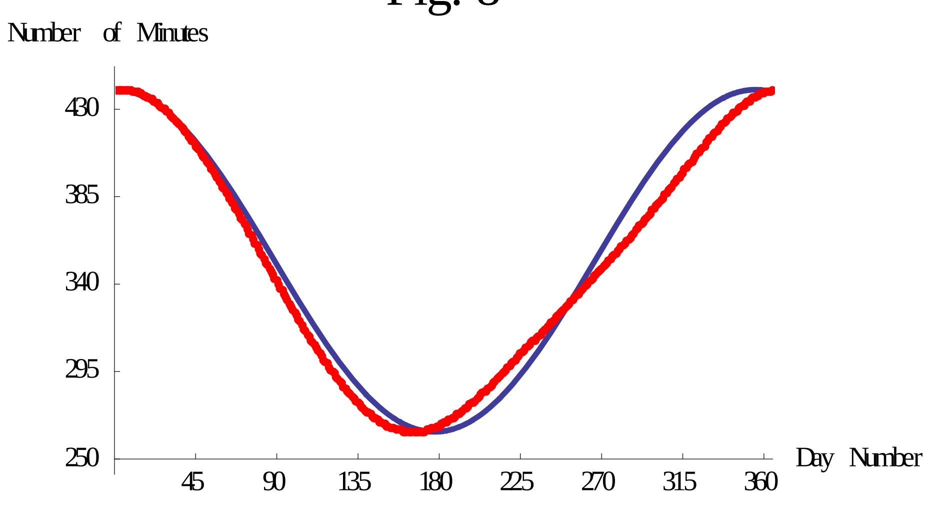
After that, we need to modify the period of the function. Due to how the period is already 360, and how the length of one year is 365 days, we know that we do not need to make major modifications. However, some tweeking is in order. To modify this, we can divide the current period of 365 by the previous period of 360 to get our B value as 1.01389.See Fig. 7.

Fig. 7



Finally, we have achieved an equation that satisfies our requirements! Our equation is as follows: $t(x) = 88\cos[(1.01389)x] + 352$. Does this equation work as we have hoped? Let us show both this function as well as the actual data: See Fig. 8.

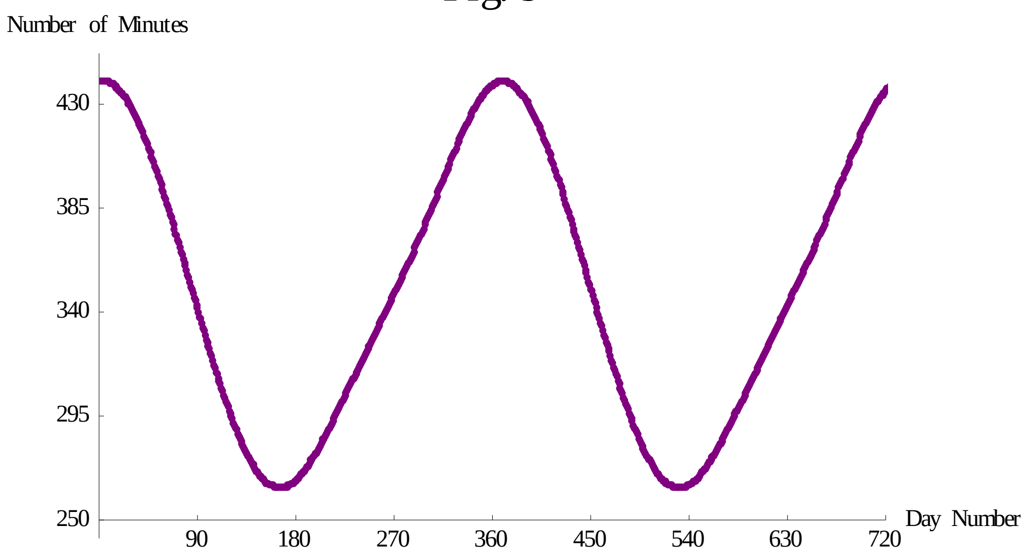
Fig. 8



Actual Points	Red
Regression	Blue

Before we go on congratulating our success, let us compare how our function works for other years. Even though this function does seem to fit the sunrise times in New York in the year 2003, would it be able to fit the sunrise times of the year 2004? What about 2005? Using the above stated technique to get data, we get the following as the plot of our new data: See Fig 9.

Fig. 9



As we add on an overlay of our function, we see that it still fits. This helps us believe that we have found the correct function. However, could our computer help us find an even more exact function? Using the nonlinear regression tool on Mathematica, it gives us the following function. See Fig. 10 and Fig. 11.

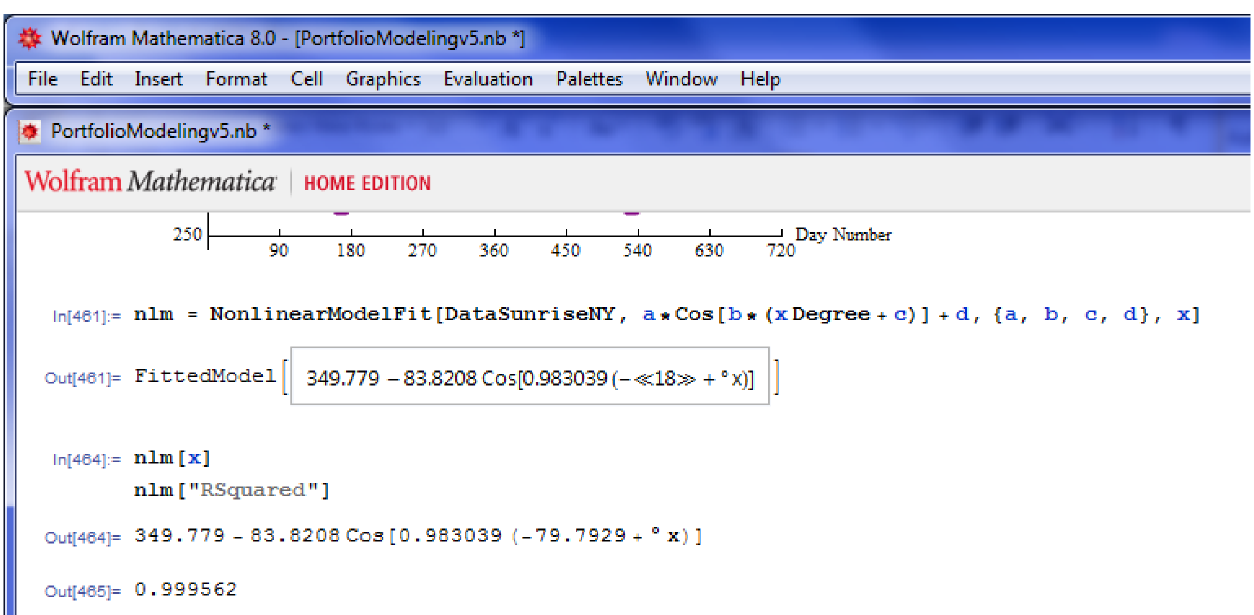
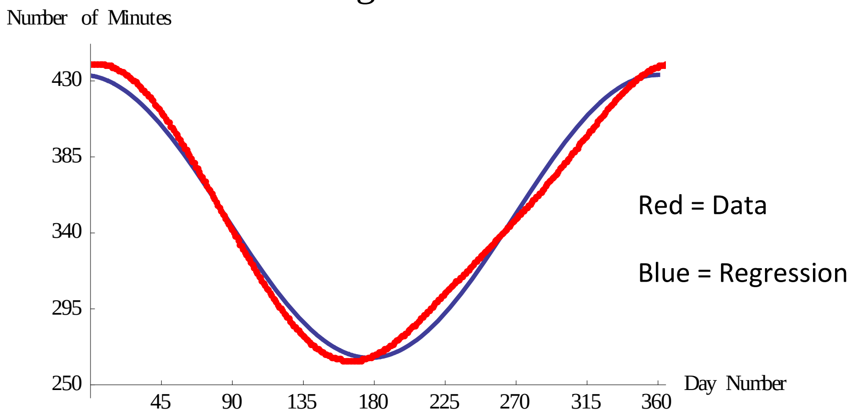


Fig. 11



As we look at our function, we notice that there are still anomalies. For instance, the majority of our information does not perfectly match up with our regression function, as we can see from before. The best way to determine just the accuracy of your regression is by finding the R^2 value of it. Before we do so, what exactly is a R^2 value, and what does it help you do? The R^2 value is a statistical term to measure if you were to pick a data point from your set at random, what is the probability that it will land on your curve? The formulas for calculating R^2 are as follows:

$$R^2 = 1 - \frac{SSErr}{SSTot}$$

$$SSErr = \sum_{a}^{i} \left(y_a - f(i)\right)^2$$

$$SSTot = \sum_{a}^{i} (y_a - \bar{y})^2$$

$$\bar{y} = \frac{1}{n}\sum_{a}^{n} y_a$$

The way that we will do this is by applying a simple loop in Mathematica that will compute the difference between the actual data and our regression. Next, we will square each of the differences and then sum them. This value is the sum of the squares residual, and for our data, it is equal to 115,875. For our next step, we must find the total sum of squares. This step is very similar to the sun of squares residual, but instead of finding the difference between the projected value and the actual value, we find the difference between the actual value and the average of the actual values. By executing these steps, we get $6.2377 \cdot 10^6$. Quite a large number, isn't it? Now, we need to divide the total sum by the residual sum, giving us the answer .0440126. Finally, we subtract this value from 1, giving us a R^2 term of .955897. This R^2 value is quite accurate, as the closer it is to 1, the more accurate it is.

Now that we have this lovely function for the sunrise function, what could we use it for? Let's rephrase that question. What kind of problems would require the user to know the sunrise of certain days throughout the year? Well, aren't we doing this model in order to solve the streetlight problem? Of course, we can use this data to fix the streetlamp system that help keeps us safe and the streets lighted. What if we were able to write a program that could shut off the streetlights automatically? A program like this would be able to greatly improve efficiency within the city, as well as save some much needed money in this recession. This program would be simple to create and simple to execute, as all it does is check on the date, find the day of the year, and then gets a …number of minutes? How are we going to be able to use a number of minutes to determine when to shut off the streetlights?

In order to really make our above equation useful, we must also create an equation that allows us to easily convert from number of minutes to a military time. From that, we would be able to program the computers to read the time. How would this function work? First, we need to determine how many hours are in the total minutes. Therefore, we need to divide the total number of minutes by sixty and disregard everything after the decimal point. We use a similar method to the way we found the original function. Therefore, the number of hours is $Floor\left[\frac{x}{60}\right]$. The number of minutes is therefore just the number of hours multiplied by sixty subtracted from the original amount. Algebraically, this is $x - Floor\left[\frac{x}{60}\right] * 60$. Now, we combine the two numbers by multiplying the number of hours by 100 and adding it to the number of minutes, resulting in the final formula: $b[x] = Floor\left[\frac{x}{60}\right] * 100 + x - (Floor\left[\frac{x}{60}\right] * 60)$. Using this, we can apply it to $t(x)$in order to get a time that is readable.

In order to make sure this little project of ours works, let us test a couple of points.

(Fig. 12)

Day number	Function results	Actual sunrise
1	720	720
90	550	542
252	530	530
360	720	718
23	713	713
145	438	431
180.5	454	n/a
-50	648	638
380	712	718

As we can see, there are many anomalies within our program, which we can now exclude as restrictions. For instance, when we tested the point 180.5, our program returned us a number. While this number is close to the point of 180, in reality, there is no sunrise for the 180.5th day of the year. In fact, that day doesn't even exist! Therefore, we can conclude that as a restriction, all of our inputs must be integers. We also tested the cases of using negative numbers and numbers over 365, which is over a single year. However, in both cases, the program worked. While the data found was not within the year 2003, it could be found in the data sets of 2002 and 2004. Therefore, our restrictions are only that x is within the set of integers.

After implementation of this simple little program, the city will be able to have streetlamps that shut down automatically each and every day, saving money along the way. What are some other ways that we could use this formula to aid the citizens of this world? Suppose there was a runner prepping to run a marathon. For now, let's just call him Robert. Now, Robert would like to wake up at the crack of dawn to run for a while, because he has to get to school at

7:30 every day, but he isn't allowed to run before the sun rises. Because the sun changes the time that it rises every day, he is really frustrated with calculating when to set his alarm clock every day. However, if he would download this simple app that we create using this function, he would get a timely wakeup call every day, which he can set up with his own favorite song blasting away! Of course, this isn't exactly free, but we will be giving all the proceeds made to the city to develop better streetlights!

Unfortunately, through all of our calculations, we have omitted a tiny feature, and suddenly, Robert's calling us, screaming that he's being waked up a full hour before the sun rose. How could this be? Isn't all our math sound? See fig. 13.

Fig 13

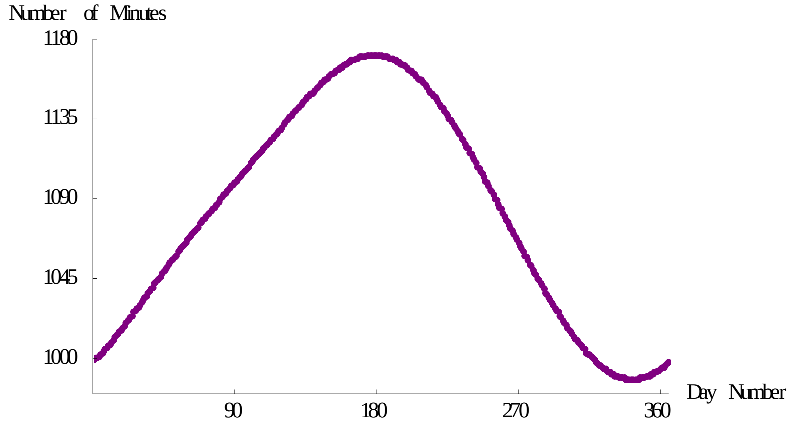
Day	Function time	Actual Time
90	550	542
180	424	527
250	527	628
300	641	620

However, there is a tiny problem. As you may see, according to our formulas, the sun rises very early, earlier than what most people can handle. Because we want for a time to represent a definitive or at least relative time, we engage in Daylight Savings Time every year, in order to reregulate the rising of the sun. While we obviously see why people would like to do this, it poses a great inconvenience to our program, as the day that DST begins changes every year. Therefore, we can't fully solve this problem without access of some information from the internet, allowing our programs to know when DST begins so that it would be able to adjust accordingly.

After finding all of the information of when the sun rises, the city council calls us once again, requesting our help once again. Why? Apparently, the old manager of when to turn on the streetlights has moved away to Antarctica due to what he said was the "pressures of the job". Now, the city doesn't know when to turn on the streetlights, and the citizens aren't feeling safe. How would we be able to create a program to automatically turn on the lights?

In order to design this program, we must process the problem as how to find a sunset formula. The basics of this formula are the same as the sunrise formula: We still need to figure out a function to model this, as well as the transformations to fit our data. Because the sun rises in a sinusoidal way, could we assume that it sets in a sinusoidal way as well? Not exactly. While the sun may set in a manner that is similar to a sine function, does it behave more like a sine function or the related cosine function? Let's take a look at the graph and data. See Fig. 14.

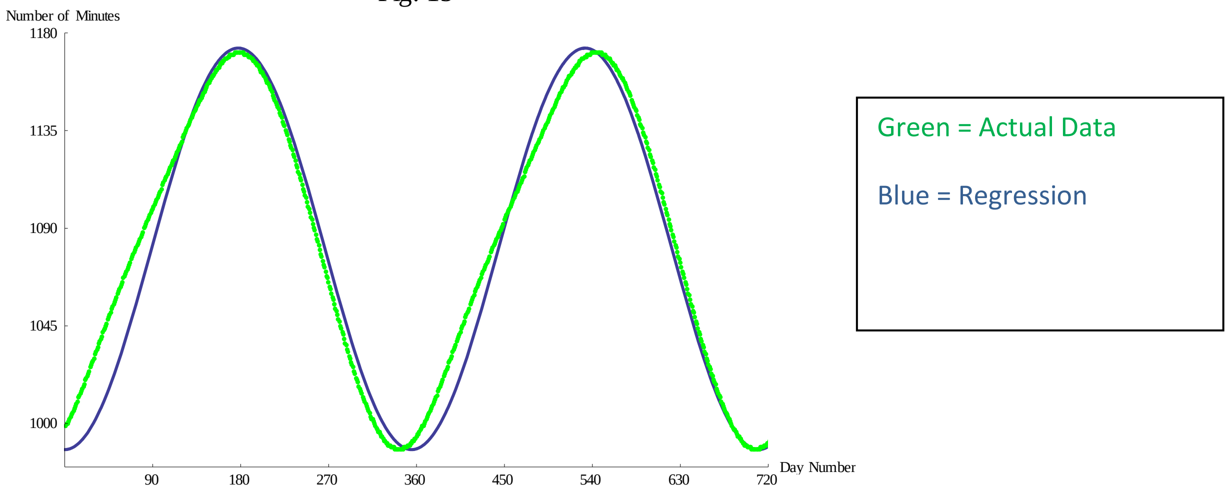
Fig. 14



From our data, we can see that the highest point for this equation is closer to the middle than to the ends of the period. However, there isn't a single trigonometric function that begins at the low point, rises to the high point, and then goes back down. What should we do? Do we need to invent a "new" function? Not quite. First, do we see any other trigonometric functions that begins at an extreme, move to another extreme, and then comes back down? Hint: we just looked at it! That is correct, this new data does resemble the cosine function, but instead of starting at a maximum, we are beginning with the minimum. How can we fix this? One simple way is to not use the $\cos(x)$, but to use $-\cos(x)$. Now that we have our function, let us analyze the transformations.

As before, our final formula must be in the form of $y = -A\cos\big(B(x-k)\big) + h$. The first thing that we should do is find the high points and low points, (crest and trough), and thus find the double amplitude. The crest is at 1171 minutes, while the trough is at 988 minutes, and by finding the difference, we find that the double amplitude is at 183 minutes. Therefore, A must be 92.5, as 92.5 is half of 183. The next part that we need to find is the vertical shift, or the h term. We notice that the trough plus the amplitude is equal to the h term, and therefore, the h term is equivalent to 988 + 92.5, or 1080.5. Finally, we shall find the B term, which is again found by taking current period divided by 360, or the original period of the function. Therefore, B = $\frac{365}{360}$. Plugging all of these values back in, we get a final equation of $s(x) = -92.5 * \cos\left(\frac{365}{360} \cdot x\right) + 1080.5$. Congratulations, we have finished with the regression of the sunset. Now, let us see this regression and overlay it with the data points. For added accuracy, let's add in the data from the year 2004, to see if this pattern continues. See Figure 15.

Fig. 15



As we did for the sunrise equation, we must also determine the accuracy of this equation, and find the R^2value. This time, we won't go as deep for explanation, but instead, list the $SSErr, SSTot, and\ R^2$ values.$SSErr$: 114119, $SSTot$: $2.62377 \cdot 10^6$. $\frac{SSErr}{SSTot}$: .0433457. $R^2 =$ .95665. Again, we have very high R^2 values, proving that our regression is accurate.

Now that we have found this formula, it is just as easy to create a program that will calculate when to turn on the lights. In fact, we follow the exact same procedure that we used to create the sunrise formula, substituting our sunset formula s(x)for the sunrise formula $t(x)$. While we have the same problem with the Daylight Saving time, we can solve it in the same way, by allowing for the internet to search up when it starts. Now, we have finished for the city of New York to automatize its streetlamp program and increase its efficiency.

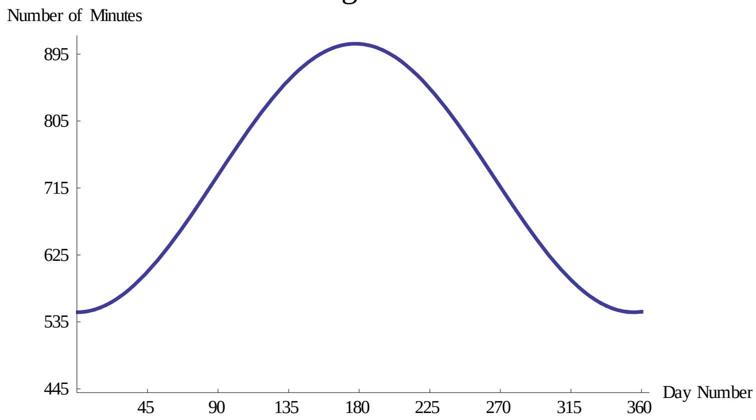
Suppose that the army academy is trying to figure out how many hours of daylight is there per day, due to a silly federal law stating that the only times you could practice parachuting was during days that are longer than 12 hours. They also believe it is a silly and ridiculous law, but how were they going to argue with the people in Congress? Now, they have tasked us with coming up a program to notify the colonels when they can carry out this practice. How can we do this?

First, we must understand what they are asking us to do. Essentially, by trying to find the days that are more than 12 hours long, we must find the length of day of each and every day. How could we do that? Let's first consider the definition of how long a day is. If you take it literally, every day is just 24 hours long, but in our context, it is the point from which the sun rises to the point where the sun sets. Hmmm, doesn't that sound familiar? Didn't we just make two equations to model when the sunrise is and when the sunset is? How could we use these two equations to find an equation for the length of day?

One method that we can try is to find the difference between the time of sunset and the time of sunrise. As this pertains to our definition of how long the day is, it works. Therefore, our equation is $j(x) = -92.5 * \cos\left(\frac{365}{360} \cdot x\right) + 1080.5 - (88\cos(\frac{365}{360} \cdot x) + 352)$. Now, is there any way to simplify this formula? The obvious elimination pertains, and we can immediately subtract 352 from 1080.5, leaving us with the vertical shift as 728.5. Now, we are left with $j(x) = -92.5 \cdot \cos\left(\frac{365}{360} \cdot x\right) - 88\cos\left(\frac{365}{360} \cdot x\right) + 728.5$. A trick that we can do is treat $\cos\left(\frac{365}{360} \cdot x\right)$ as a unit by itself, leaving us with $-92.5a - 88a$, or $-180.5a$. Substituting back

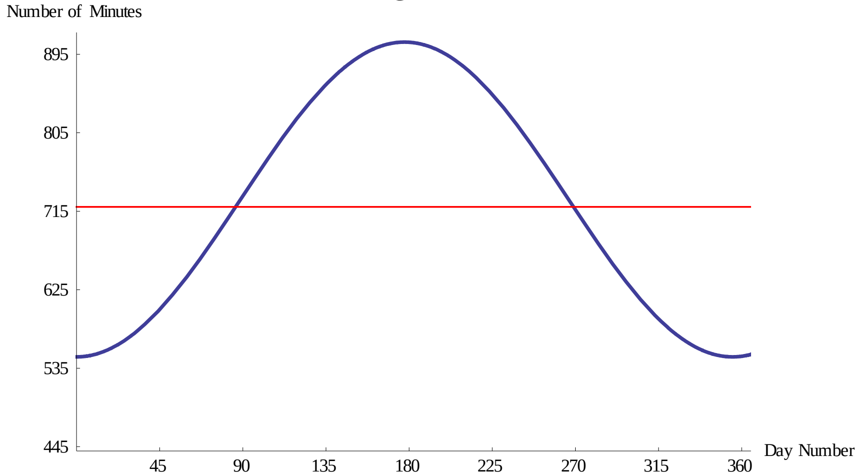
$\cos\left(\frac{365}{360}\cdot x\right)$for a, we are left with the final formula of $-180.5\cdot\cos\left(\frac{365}{360}\cdot x\right)+728.5$. Let us get a visual representation of this: See Figure 16.

Fig. 16



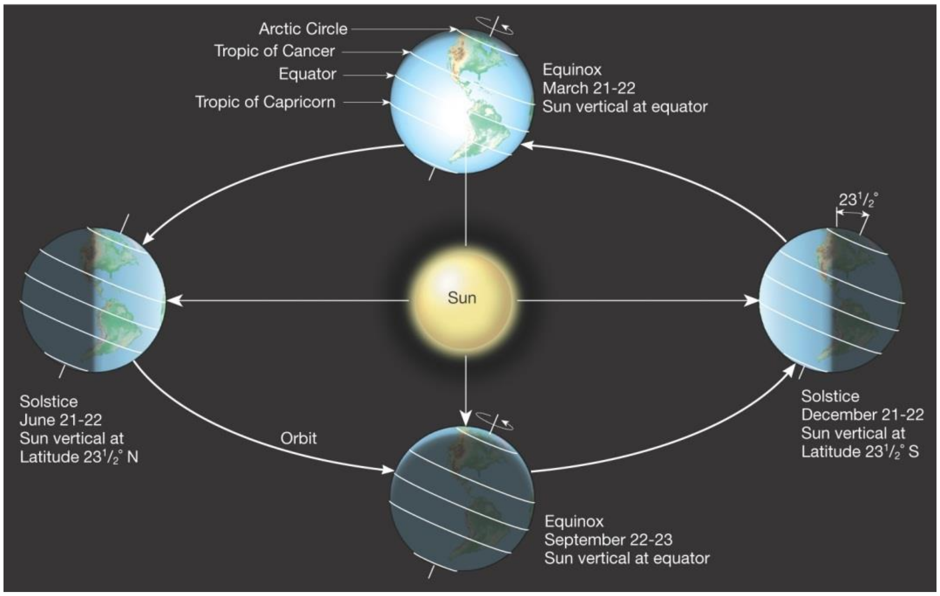
Now that we have this information, how are we going to solve the Army's problems? It is quite simple, really. All we have to do is find the intersection of the line $y = 720$ and our equation, and we have the range of days that are greater than 12 hours long. See Fig. 17.

Fig. 17



In the year 2003, this means that the days between the 87th day and the 267th day were longer than 12 hours. After a couple of hours of programming in Java, the Army's top ranking officials have this cute little app that tells them when they can execute these exercises.

A little sidenote here, there is something rather significant about these two dates. First, as you probably have noticed, is that these days have equal amounts of daylight and night. There is 12 hours of daylight and 12 hours of sunlight. Secondly, the days that these special events fall on just so happen to divide the year up into 4ths. In fact, these 2 days are generally known as the Spring and Autumn equinoxes, generally around the dates of March 21st and November 21st respectively. For further reference on why these days have even daylight, refer to this diagram:

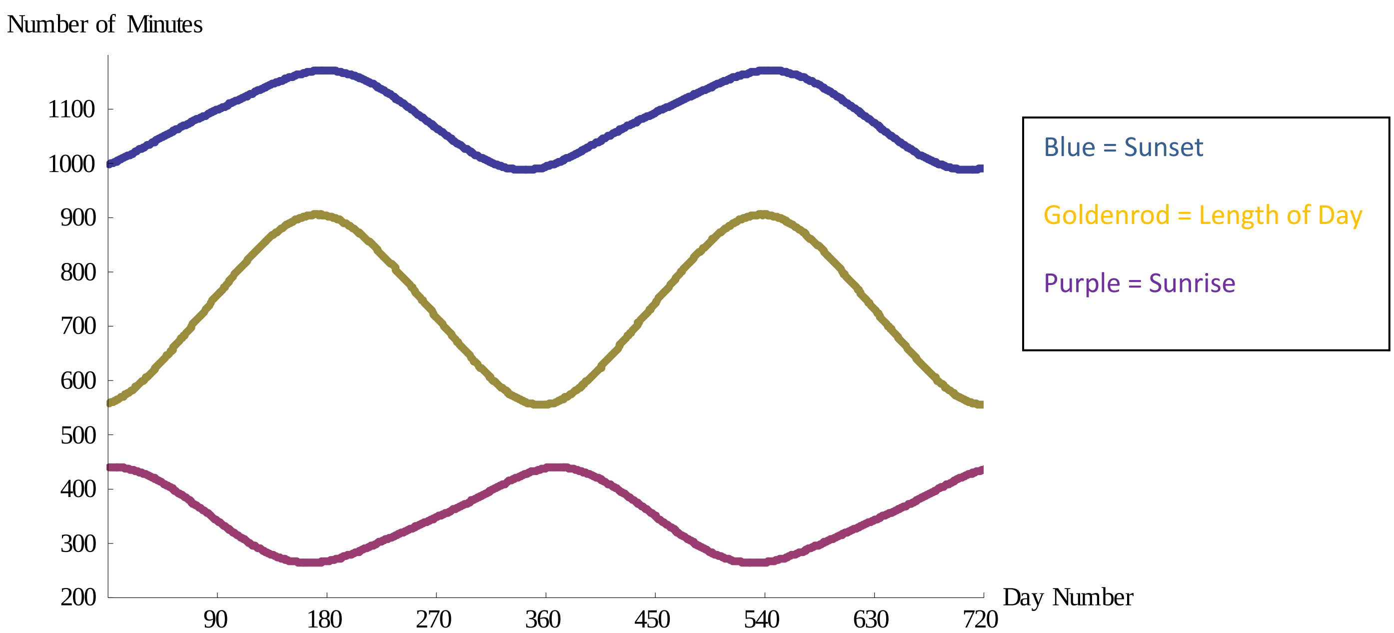


As you can see, all of this has to deal with the tilt of the earth and its relativeness to the sun. However, this goes further into astronomy, which for the sake of time, we won't delve too far into here.

After we have finished our contacts with the Army, a new society in New York calls us up. Apparently, they are trying to build a modern day Stonehenge in the middle of Central Park. However, they are missing a key piece of information: the shortest day of the year. This piece of information is extremely necessary, as they need to calibrate their stones so that the sun shines through the keystone at exactly the shortest day. As you know, it is quite a pain to move huge 20 ton rocks, and they really want to get it right. How could we find the shortest day of the year?

This is actually a very simple question, as due to the nature the cosine function, we know that the shortest day of the year is also the trough of the function. In this case, we know that the negative cosine function has its trough on the first day of the year. Simple, right? Not exactly. Not only does the society want to know the day of the shortest day, but also exactly how long it is. This again is a quite easy task, as all we need to do is plug in 1 as our x value, and our function does the rest for us. The result is a day lasting 548 minutes long, and using our $g(x)$ function, we find that equal to 9 hours and 8 minutes.

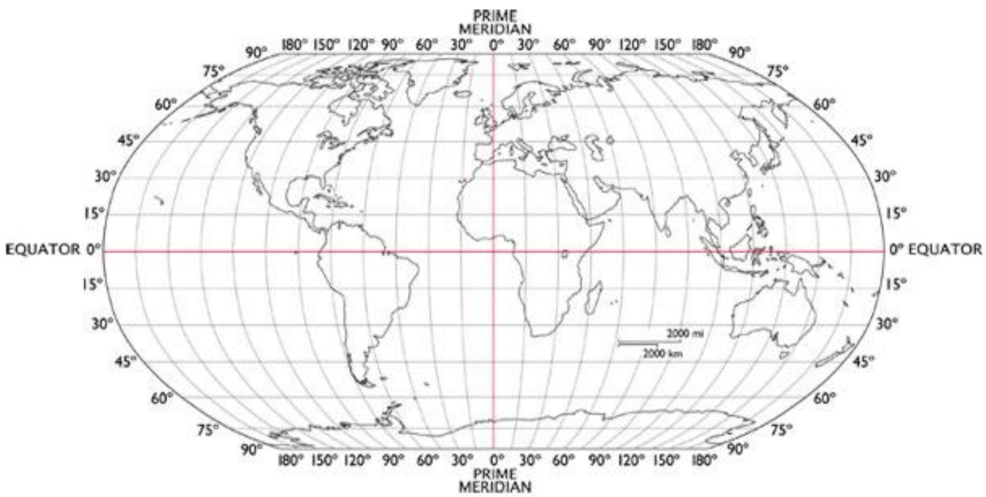
Now that we have all of our information for the city of New York City, let us compile all of the data onto one graph, just to see our progress:



As our success grows in New York City, winning medals for the most innovative way of saving energy and money, other cities also wish to have programs like this. In fact, three cities have already stood up to ask for this program: the city of Mathville, located exactly 1000 kilometers west of New York City, the city of Calculus, located exactly 1000 kilometers south of New York City, and the barren and desolate city of Statisticallville, located exactly 1000 kilometers north of New York City. Never mind that they each only have about 100 residents, they all want to get in on this great new plan. How can we help them?

Due to their exact locations away from New York City, we begin to see certain patterns of shifting or stretching our New York models. Let's start off at Mathville, 1000 kilometers west of New York. From common knowledge, we know that the sunrise should be later, as we all know about the time zones increasing as you go towards the west. Because for now, all times will be in Eastern Standard Time, we can anticipate a downwards vertical shift as the sun rises earlier. However, by how much? As we do not have much experience yet with these transformations, let us look at this cities data.

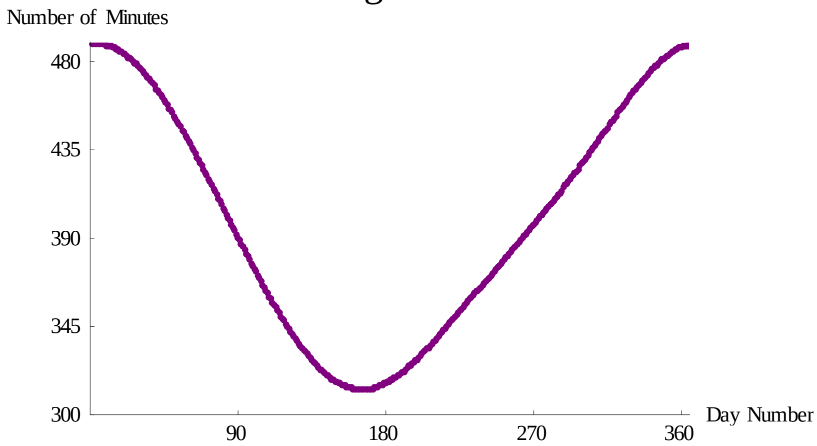
In order to find this city's data, we must first find its longitude and latitude. Just to refresh your memory, any position on Earth can be determined by a longitude and latitude



coordinate, like this map:

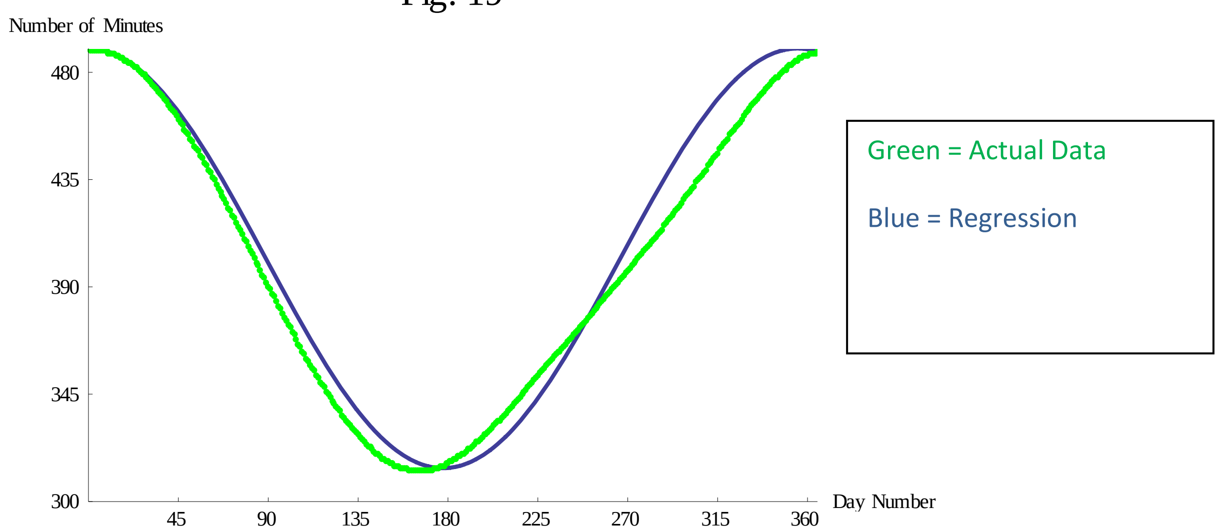
We also know how long one degree of longitude/latitude is. While it differs for different latitudes, one degree of longitude at the 40 degree mark, where New York City is, is about 85.39 kilometers. Given that New York's coordinates are $(73°55'W\ , 40°44'N)$, we can find Mathville: about $(86°10'W, 40°44'N)$ According to the Naval observatory, this is the sunrise times: See Fig. 18.

Fig. 18



This graph actually looks exactly the same as figure 4, with the only difference a shift in the vertical axis. Assuming that that is the only difference, let us find the new shift. This model's maximum is now at 489, with its minimum at 313. The difference is 176, which when we take half of it, we get 88. Wow, wasn't 88 our previous stretch factor as well? Now, to find the vertical shift, we need to add 88 to our lowest value, which is 313, to get 402. This equation is

Fig. 19



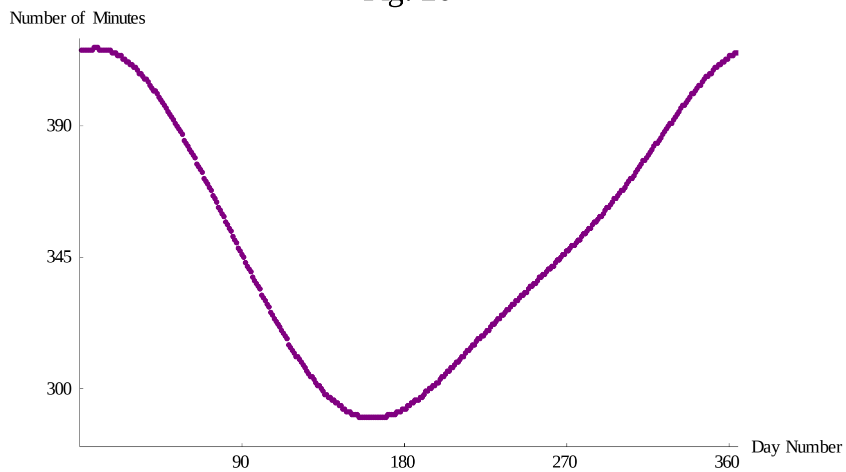
now $k(x) = 88 \cdot \cos\left(\frac{365}{360} \cdot x\right) + 402$. Let us plot this, compare, and get a R^2 value. See Fig. 19.

As you can see, it again appears to be very close fitting. We will now apply a R^2 value to this. $SSErr = 46490.4,\ SSTot = 1.29779 \cdot 10^6, R^2 = .964178$. Congratulations, we have found a successful transformation!

For our North and South movements, we must investigate some extreme cases. As you may know, at the North Pole, there is this odd phenomenon called the midnight sun, where there is day for half of the year and night for the other half of the year. Can we use the phenomenon to explain these shifts? Let's look at data.

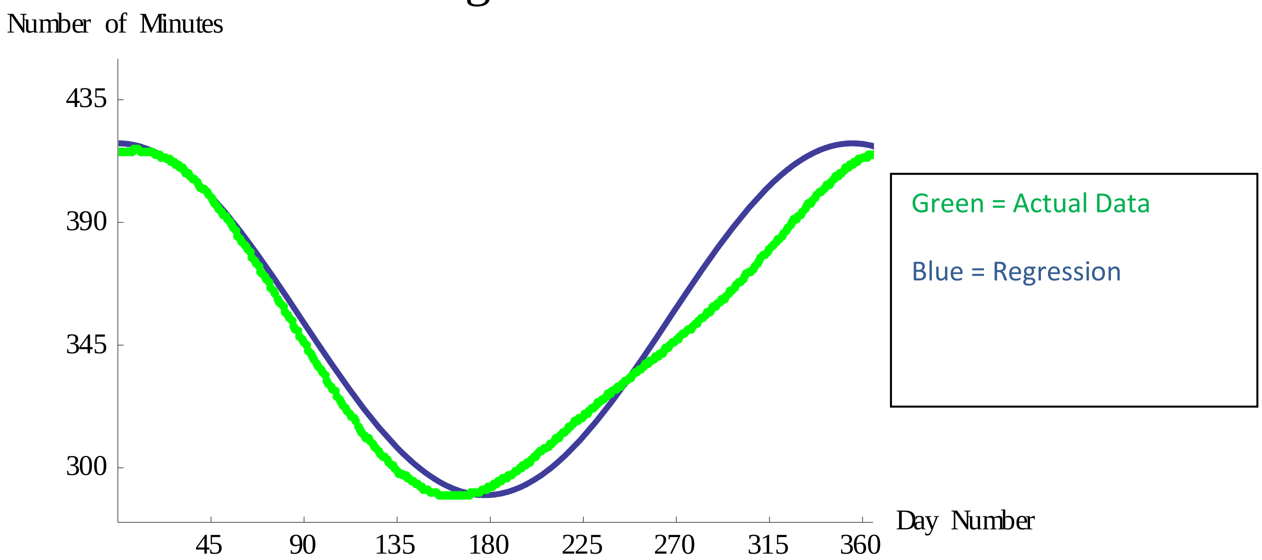
For the city of Calculus, we can calculate it's coordinates to be $(73°55'W,\ 31°44'N)$, as one degree of latitude is about 1193 kilometers. From this coordinate, we can find the following coordinates: See Figure 20.

Fig. 20



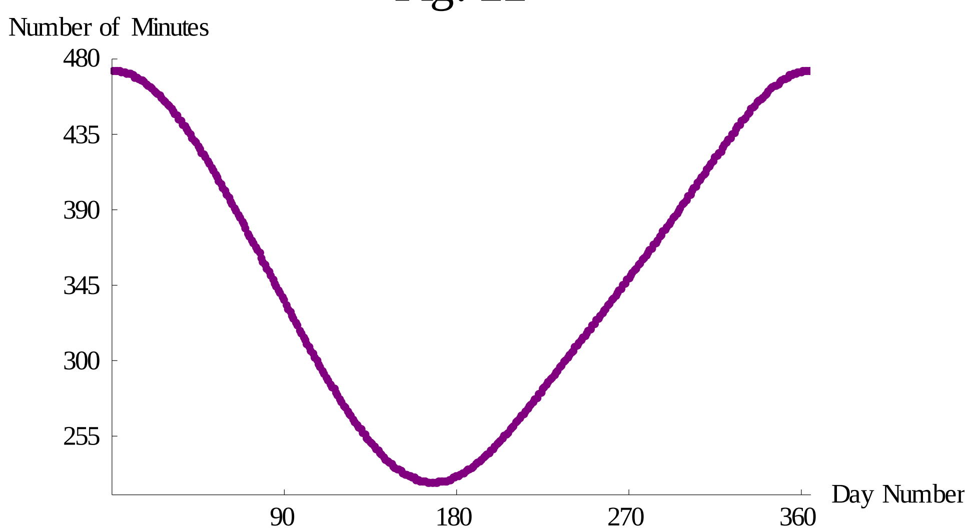
Again, we see a very similar graph. However, when we begin analyzing, we notice that some things have changed a bit more. For instance, the maximum is at 427 and the minimum is at 290, which makes the double amplitude be 137. Therefore, the new amplitude is only 64.5, significantly less than the 88 that we had previously. Our vertical shift is adjusted to be 354.5, and our equation is calculated as $m(x) = 64.5\cos\left(\frac{365}{360} \cdot x\right) + 354.5$. See figure 21.

Fig. 21



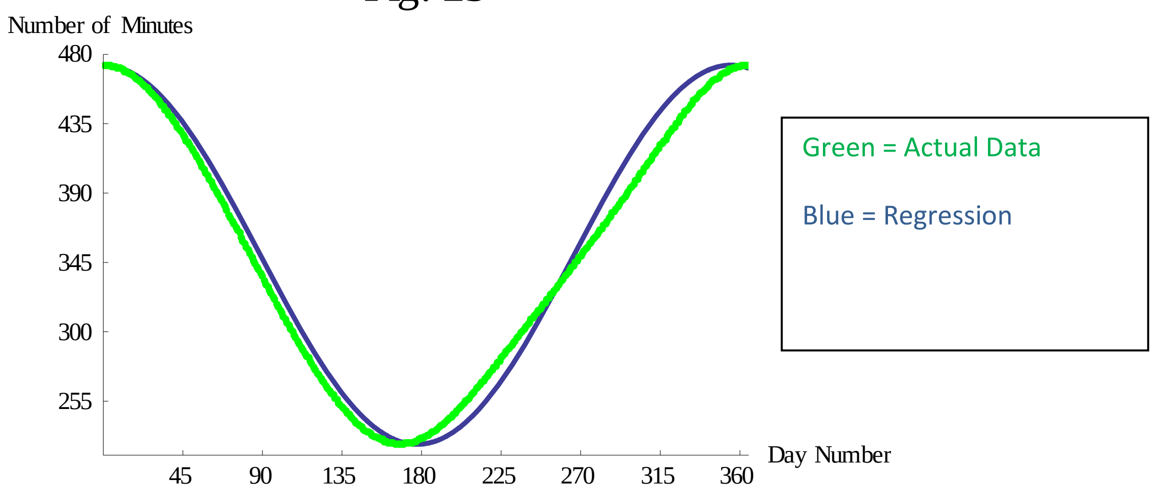
Our final city is the barren city of Statisticallville, who has the coordinates $(73°55'W, 49°44'N)$ as determined by converting between degrees of latitude to kilometers. When we request information about this city, we notice that the city's data seems to be more spread out than the previous models. Let's take a look at the plot: See Figure 22.

Fig. 22



As we see from the plot, this model has a larger amplitude than New York's, which does make sense. In order to find the exact amplitude, we find the maximum at 473 and a minimum at 227. When we find the difference, we get 246, and after we divide by 2, we get our vertical stretch as 123. As we can see, this is much larger than the previous stretch of 88. To find the vertical shift, we add our amplitude to our minimum value, and get 350. Oddly enough, this number is pretty consistent through going North, South, or just in New York. Therefore, our final formula is $n[\mathrm{x}] := 123 \cdot \mathrm{Cos}((365/360) \cdot x) + 350$. Let us take a look at this regression along with the actual data: See Figure 23.

Fig. 23



Congratulations! All of our cities are satisfied with their new programs, and we have completed the task of modeling the sunrise times in multiple cities around the US. From our studies, we have looked at how to model a sunrise function, which function is best for modeling this, how accurate our model is, and how we could practically use our model in real-world problems.