```
In[1]:= Unprotect[C, D]
Out[1]= {C, D}
```

- **Inputing city data:**

Radiation and ACEnergy Data:
The data for Radiation and ACEnergy was obtained via National Renewable Energy Laboratory (NREL) Home Page. By simply entering the City Zip Code and data was given for the city on a monthly period. Each city had data points for the month's radiation and the amount of ACEnergy produced from that Radiation. The units for Solar Radiation is (kWh/m2/day) and the units for ACEnergy is (kWh).

NEW YORK (JFK AP), NY:

```
In[2]:= City1 = NEW YORK;
In[3]:= Wind1 = {13.0, 13.3, 13.5, 12.7, 11.6, 10.7, 10.2, 10.0, 10.4, 11.0, 12.2, 12.7};
In[4]:= Radiation1 = {3.17, 4.17, 4.57, 5.25, 5.30, 5.78, 5.64, 5.52, 5.06, 4.48, 2.90, 2.85};
In[5]:= ACEnergy1 = {315, 370, 432, 466, 472, 484, 479, 475, 431, 410, 264, 276};
```

PHOENIX, AZ:

```
In[6]:= City2 = PHOENIX;
In[7]:= Wind2 = {5.3, 5.8, 6.6, 6.9, 7.0, 6.7, 7.1, 6.6, 6.3, 5.8, 5.3, 5.1};
In[8]:= Radiation2 = {5.09, 6.06, 6.61, 7.54, 7.53, 7.28, 7.13, 7.17, 7.15, 6.75, 5.59, 4.88};
In[9]:= ACEnergy2 = {451, 486, 566, 613, 619, 559, 569, 576, 557, 566, 469, 438};
```

CHICAGO, IL:

```
In[10]:= City3 = CHICAGO;
In[11]:= Wind3 = {11.6, 11.4, 11.8, 11.9, 10.5, 9.3, 8.4, 8.2, 8.9, 10.1, 11.1, 11.0};
In[12]:= Radiation3 = {3.04, 3.78, 4.34, 5.11, 5.68, 5.66, 5.92, 5.22, 4.94, 4.26, 2.83, 2.27};
In[13]:= ACEnergy3 = {302, 336, 419, 455, 498, 470, 497, 444, 414, 389, 258, 220};
```

SEATTLE (SEA-TAC AP), WA

```
In[14]:= City4 = SEATTLE;
In[15]:= Wind4 = {9.5, 9.4, 9.4, 9.4, 8.9, 8.6, 8.1, 7.8, 8.0, 8.3, 9.1, 9.6};
In[16]:= Radiation4 = {1.54, 2.50, 3.71, 4.37, 5.31, 5.52, 5.88, 5.17, 4.98, 3.00, 1.76, 1.26};
In[17]:= ACEnergy4 = {133, 201, 335, 383, 471, 467, 508, 448, 419, 263, 148, 103};
```

In[18]:= $\text{Table}\left[\text{EnergyValue}_i = \text{ACEnergy}_i * E_{cost} * \frac{\text{dollars}}{100\ \text{cents}}, \{i, 1, 4\}\right];$

Obtaining a function for the curves of Radiation.

The first step in analyzing the amount of energy a city can obtain from a solar panel is to analyze the amout of radiation a city recieved throughout the year. Below is a plot of the solar radiation of a city as a function of month.

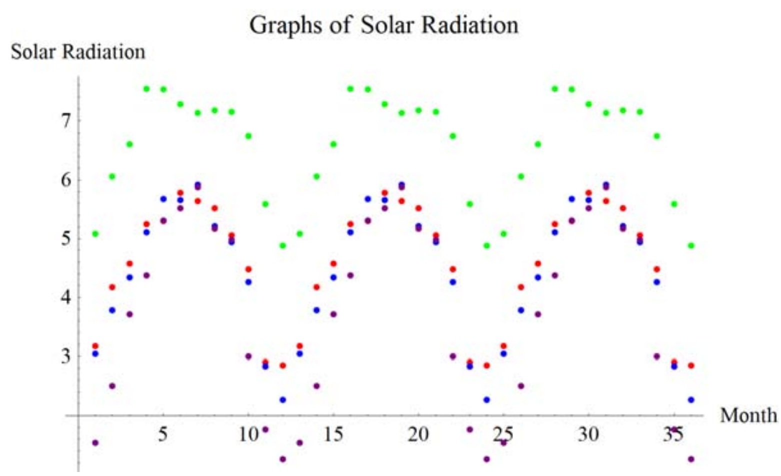
Reinput Radiation Data for this specific calculation:

```
In[19]:= RadiationNew1 = {3.17, 4.17, 4.57, 5.25, 5.30, 5.78, 5.64, 5.52, 5.06, 4.48,
            2.90, 2.85, 3.17, 4.17, 4.57, 5.25, 5.30, 5.78, 5.64, 5.52, 5.06, 4.48, 2.90,
            2.85, 3.17, 4.17, 4.57, 5.25, 5.30, 5.78, 5.64, 5.52, 5.06, 4.48, 2.90, 2.85};
         RadiationNew2 = {5.09, 6.06, 6.61, 7.54, 7.53, 7.28, 7.13, 7.17, 7.15, 6.75, 5.59,
            4.88, 5.09, 6.06, 6.61, 7.54, 7.53, 7.28, 7.13, 7.17, 7.15, 6.75, 5.59, 4.88,
            5.09, 6.06, 6.61, 7.54, 7.53, 7.28, 7.13, 7.17, 7.15, 6.75, 5.59, 4.88};
         RadiationNew3 = {3.04, 3.78, 4.34, 5.11, 5.68, 5.66, 5.92, 5.22, 4.94, 4.26, 2.83,
            2.27, 3.04, 3.78, 4.34, 5.11, 5.68, 5.66, 5.92, 5.22, 4.94, 4.26, 2.83, 2.27,
            3.04, 3.78, 4.34, 5.11, 5.68, 5.66, 5.92, 5.22, 4.94, 4.26, 2.83, 2.27};
         RadiationNew4 = {1.54, 2.50, 3.71, 4.37, 5.31, 5.52, 5.88, 5.17, 4.98, 3.00, 1.76,
            1.26, 1.54, 2.50, 3.71, 4.37, 5.31, 5.52, 5.88, 5.17, 4.98, 3.00, 1.76, 1.26,
            1.54, 2.50, 3.71, 4.37, 5.31, 5.52, 5.88, 5.17, 4.98, 3.00, 1.76, 1.26};
```

```
In[23]:= SolarRadiationGraph1 = ListPlot[
            Table[Rdata_i = Table[{j, RadiationNew_i[[j]]}, {j, 1, Length[RadiationNew_i]}], {i, 1, 4}],
            PlotStyle → {Red, Green, Blue, Purple}, AxesLabel → {"Month", "Solar Radiation"},
            PlotLabel → "Graphs of Solar Radiation"];
```

```
In[24]:= Rasterize[SolarRadiationGraph1, ImageResolution → 300]
```

Out[24]=



After looking at this data, we would like to obtain a formula for the cities radiation as a function of time. The data points seem to have a parabolic shape and therefore we will fit a parabola to the data points. We will use the FindFit function to obtain values for the coefficients for each city.

```
In[25]:= Table[
           tempsol_i = FindFit[Rdata_i, A_rad,i + B_rad,i Sin[C_rad,i (x - D_rad,i)],
              {{A_rad,i, 4}, {B_rad,i, 4}, {C_rad,i, .5}, {D_rad,i, 15}}, {x}], {i, 1, 4}];
         Table[{A_rad,i, B_rad,i, C_rad,i, D_rad,i} = {A_rad,i, B_rad,i, C_rad,i, D_rad,i} /. tempsol_i, {i, 1, 4}];
```

After obtianing the constants, we can define the functions below. RadFunction is a function of the predicted amount of radiation a city may have at any given point in the year.

```
In[27]:= RadFunction1[x_] := Arad,1 + Brad,1 Sin[Crad,1 (x - Drad,1)]
         RadFunction2[x_] := Arad,2 + Brad,2 Sin[Crad,2 (x - Drad,2)]
         RadFunction3[x_] := Arad,3 + Brad,3 Sin[Crad,3 (x - Drad,3)]
         RadFunction4[x_] := Arad,4 + Brad,4 Sin[Crad,4 (x - Drad,4)]
```

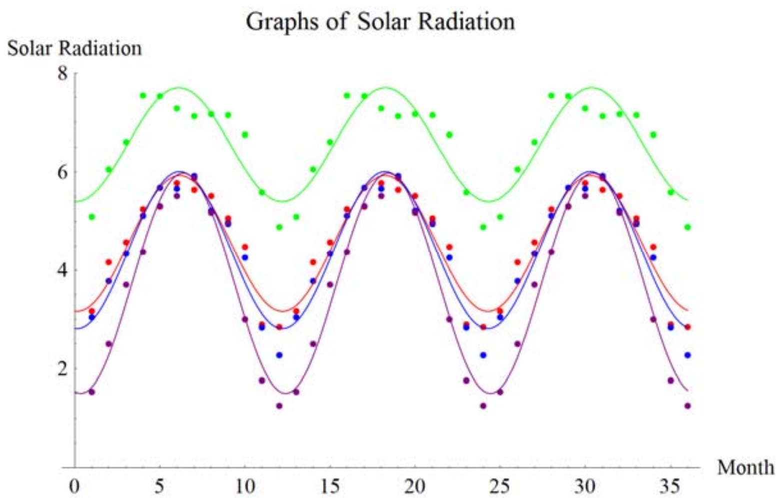
```
In[31]:= {Color1, Color2, Color3, Color4} = {Red, Green, Blue, Purple};
```

```
In[32]:= SolarRadiationGraph2 =
          Show[Table[
            Plot[RadFunctioni[x], {x, 0, 36}, PlotStyle -> {Colori}, PlotRange → {0, 8}], {i, 1, 4}]];
```

We can check our Functions by plotting them with the data points. As you may notice, the functions obtained fit the data points fairly nicely.

```
In[33]:= Rasterize[Show[SolarRadiationGraph1, SolarRadiationGraph2,
          PlotRange → {0, 8}, AxesLabel → {"Month", "Solar Radiation"},
          PlotLabel → "Graphs of Solar Radiation"], ImageResolution → 300]
```

Out[33]=



■ Obtaining a Function of Energy Produced by a solar panel.

We must obtain a function which converts solar radiation into actual energy produced. We can do this by the following method:

1. Grouping all the city data into one large set of data points. This is done simply to obtain one set of data points which *Mathematica* and use to fit a single function to the conversion of radiation into the energy produced.

```
In[34]:= Radiationtotal = Flatten[Table[
            Table[{
              Radiationi[[j]]}, {j, 1, Length[Radiationi]}
            ], {i, 1, 4}]];
```

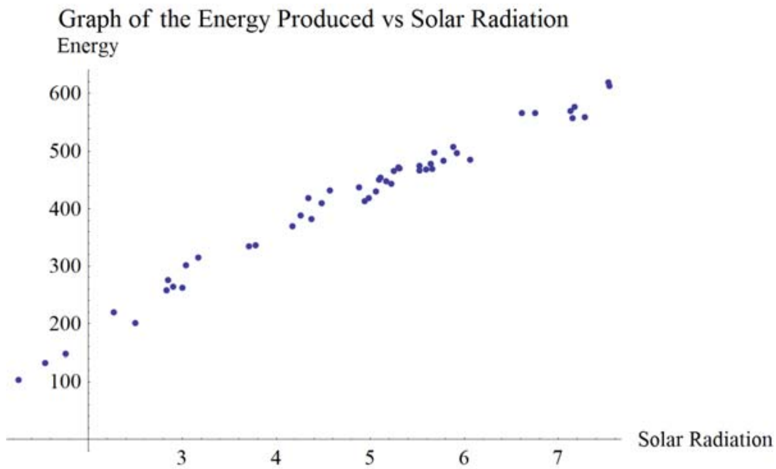
```
In[35]:= ACEnergytotal = Flatten[Table[
            Table[{
              ACEnergyi[[j]]}, {j, 1, Length[ACEnergyi]}
            ], {i, 1, 4}]];
```

We can plot the data points in order to view how the data points behave. This is done in order to see what type of function we would like to fit to the desired function.

In[36]:=
```
Points_{sun vs energy, merged} = ListPlot[
    LightPower = Table[{Radiation_total[[j]], ACEnergy_total[[j]]}, {j, 1, Length[ACEnergy_total]}],
    AxesLabel → {"Solar Radiation", "Energy"},
    PlotLabel → "Graph of the Energy Produced vs Solar Radiation"];
```

In[37]:=
```
Rasterize[Points_{sun vs energy, merged}, ImageResolution → 300]
```

Out[37]=



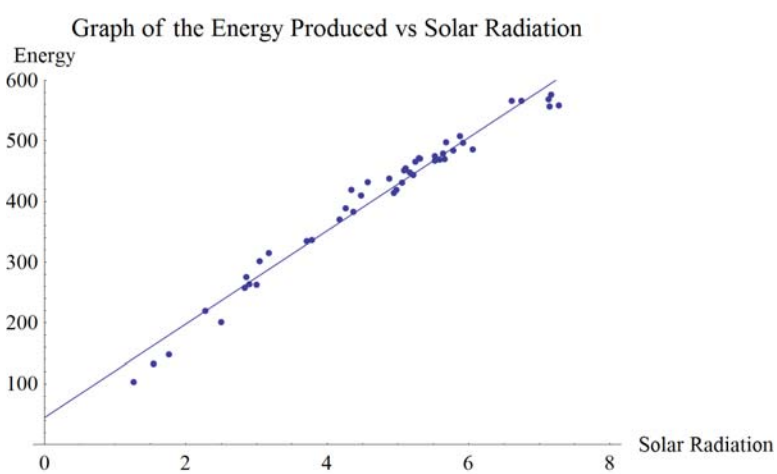
The data points look linear. We can now fit a linear function to the data points and plot the function in order to see if the constants are justifiable.

In[38]:=
```
sol_solar = FindFit[LightPower, a_solar x + b_solar, {a_solar, b_solar}, {x}];
{a_solar, b_solar} = {a_solar, b_solar} /. sol_solar;
SolarFunction[x_] := a_solar x + b_solar;
```

In[41]:=
```
sol_solarcombined = Show[Plot[SolarFunction[x], {x, 0, 8}, PlotRange → {0, 600}],
    Points_{sun vs energy, merged}, AxesLabel → {"Solar Radiation", "Energy"},
    PlotLabel → "Graph of the Energy Produced vs Solar Radiation"];
```

In[42]:=
```
Rasterize[sol_solarcombined, ImageResolution → 300]
```

Out[42]=



After plotting the function with the data points, we can conclude that its a fairly good fit and we can now proceed to the next step of analyzing data.

Temperature Data for the Cities:

We must obtain formulas for the temperatures of the 4 different citys in order to calculate the efficiency of the solar panel. This is because the efficiency of a solar panel depends on the surrounding temperatures. After obtaining formulas of temperature with respect to time of year, we can calculate the efficiency of a solar panel at that particular time of year.

Step 1: Importing all temperature data from the 4 cities.

```
In[43]:= AirportCode = {KJFK, KSEA, KPHX, KORD};
```

New York City Temperatures

```
In[44]:= << "Units`";
Tempdata1 = WeatherData["KJFK", "MeanTemperature",
   {{2006, 1, 1}, {2008, 12, 31}, "Week"}, "DateValue"];
Tempdata21 = Table[{(DateDifference[{2006, 1, 1}, Tempdata1[[i, 1]]] / 365) ,
    ConvertTemperature[Tempdata1[[i, 2]], Celsius, Fahrenheit]}, {i, Length[Tempdata1]}];
```

Seattle Temperatures

```
In[47]:= << "Units`";
Tempdata2 = WeatherData["KSEA", "MeanTemperature",
   {{2006, 1, 1}, {2008, 12, 31}, "Week"}, "DateValue"];
Tempdata22 = Table[{(DateDifference[{2006, 1, 1}, Tempdata2[[i, 1]]] / 365) ,
    ConvertTemperature[Tempdata2[[i, 2]], Celsius, Fahrenheit]}, {i, Length[Tempdata2]}];
```

Arizona Temperatures

```
In[50]:= << "Units`";
Tempdata3 = WeatherData["KPHX", "MeanTemperature",
   {{2006, 1, 1}, {2008, 12, 31}, "Week"}, "DateValue"];
Tempdata23 = Table[{(DateDifference[{2006, 1, 1}, Tempdata3[[i, 1]]] / 365) ,
    ConvertTemperature[Tempdata3[[i, 2]], Celsius, Fahrenheit]}, {i, Length[Tempdata3]}];
```

Chicago Temperatures

```
In[53]:= << "Units`";
Tempdata4 = WeatherData["KORD", "MeanTemperature",
   {{2006, 1, 1}, {2008, 12, 31}, "Week"}, "DateValue"];
Tempdata24 = Table[{(DateDifference[{2006, 1, 1}, Tempdata4[[i, 1]]] / 365) ,
    ConvertTemperature[Tempdata4[[i, 2]], Celsius, Fahrenheit]}, {i, Length[Tempdata4]}];
```

```
In[56]:= Unprotect[C, D];
```

It is known from past experience that the function of temperature with respect to time is a Sin function. Therefore, the temperature functions will be fit to the generic formula A+B Sin[C(x+D)].

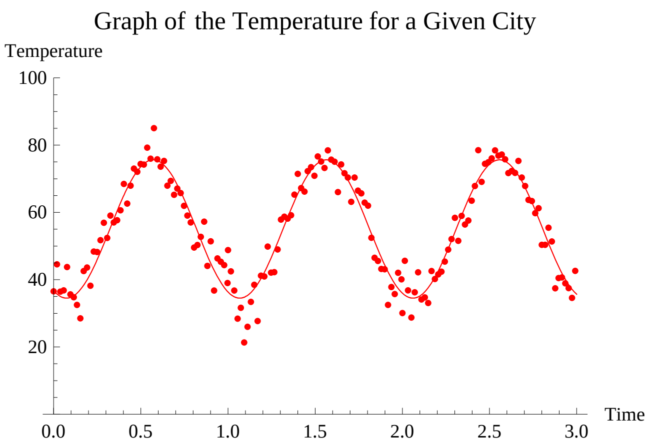
```
In[57]:= Table[
  soli = FindFit[Tempdata2i, Ai + Bi Sin[Ci (x - Di)],
    {{Ai, 50}, {Bi, 20}, {Ci, -6}, {Di, -2}}, {x}], {i, 1, 4}];
Table[{Ai, Bi, Ci, Di} = {Ai, Bi, Ci, Di} /. soli, {i, 1, 4}];
```

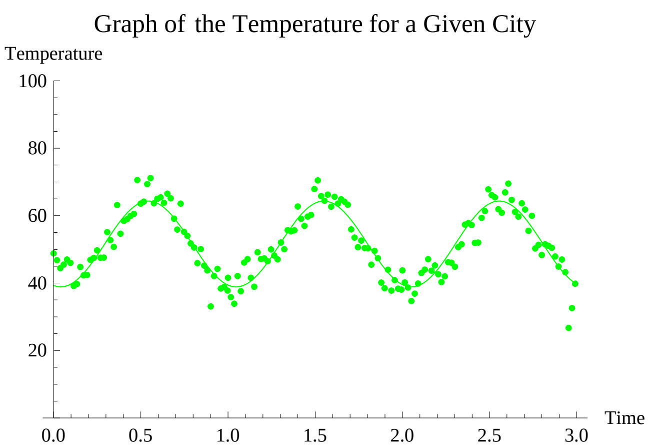
The new constants are now used to define the temperature function for a specific city, as seen below:

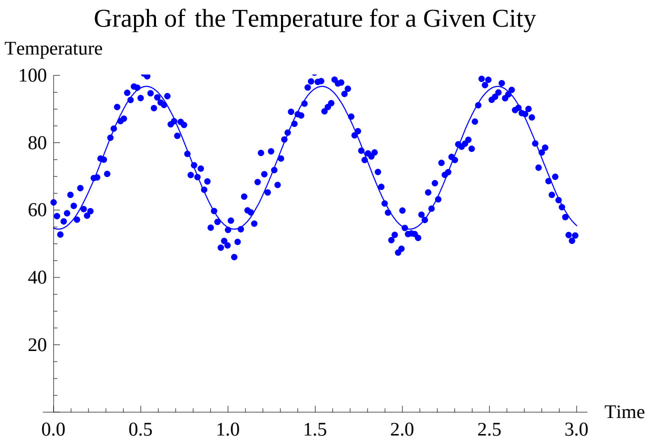
In[59]:=
```
Temperature1[x_] := A1 + B1 Sin[C1 (x - D1)]
Temperature2[x_] := A2 + B2 Sin[C2 (x - D2)]
Temperature3[x_] := A3 + B3 Sin[C3 (x - D3)]
Temperature4[x_] := A4 + B4 Sin[C4 (x - D4)]
```

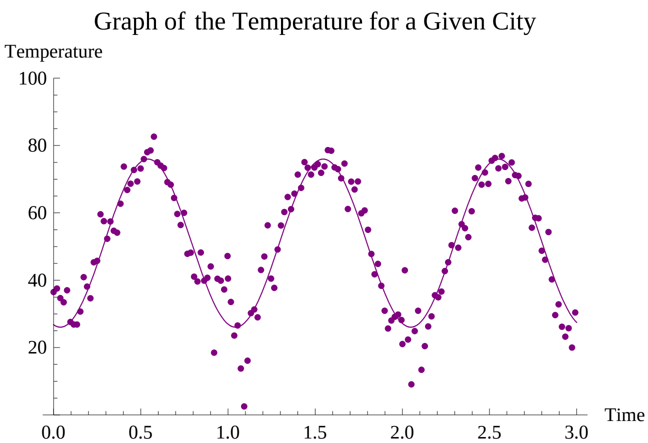
To ensure the functions obtained fit the data points accordinly, the function is graphed along with the data points.

In[63]:=
```
For[i = 1, i ≤ 4, i++,
  Print[
   Show[
    ListPlot[Tempdata2i, PlotStyle -> {Colori}],
    Plot[Ai + Bi Sin[Ci (x - Di)], {x, 0, 3}, PlotStyle → {Colori}], PlotRange → {0, 100}],
    AxesLabel → {"Time", "Temperature"},
    PlotLabel → "Graph of the Temperature for a Given City"]]];
```









Efficiency Function

The efficiency of a solar panel is a linear function which was found to be Efficiency = -4/9 T + 82. This equation expressed the efficiency out of 100 rather than 1, and therefore the equation must be divided by 100. Also, the efficiency is a function of temperature in degrees Celcius and therefore we must convert the function of temperature obtained previously from Farenheith to Celcius. The resulting equation will look as follows:

```
In[64]:= Efficiency[x_] := (- (4 / 9) * ((x - 32) / 1.8) + 82) / 100;
```

Power generated by a solar panel:

We can now combine all our formulas.

$N_{\text{solarpanels}}$ is defined to be the number of solar panels. It will be defined as 1 in order to compare the power generated by each region.

The Solar Function is used to convert radiation into the numerical value of kWhrs produced, we must use the radiation function within the solar function in order to obtain a value for radiation which can be used in order to obtain a value for power generated by the solar panel.

The efficiency fucntion is dependant on the temperature function. The temperature function must first calculate the temperature of the city at a certain time and once this value is obtained it will be used to calculate the efficiency of the solar panel at that particualr time in the year.

```
In[65]:= Nsolarpanels = 1;
```

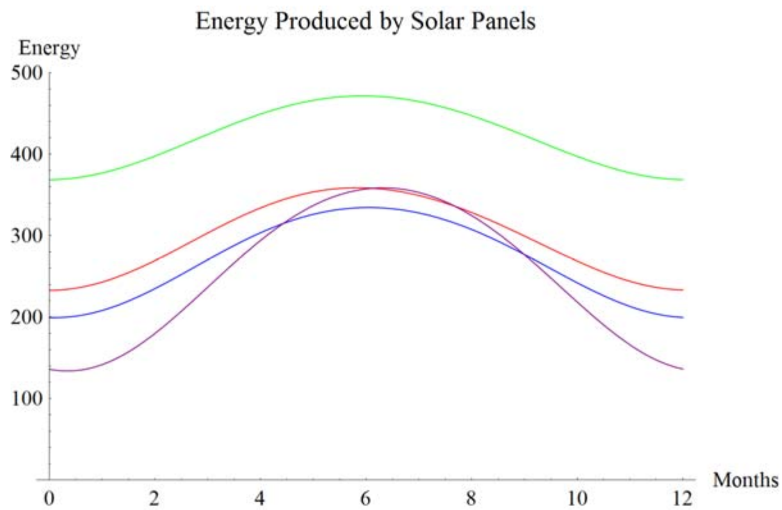
In[66]:=
```
Energy_{solar,1}[x_] := N_{solarpanels} SolarFunction[RadFunction_1[x]] * Efficiency[Temperature_1[x / 12]]
Energy_{solar,2}[x_] := N_{solarpanels} SolarFunction[RadFunction_2[x]] * Efficiency[Temperature_2[x / 12]]
Energy_{solar,3}[x_] := N_{solarpanels} SolarFunction[RadFunction_3[x]] * Efficiency[Temperature_3[x / 12]]
Energy_{solar,4}[x_] := N_{solarpanels} SolarFunction[RadFunction_4[x]] * Efficiency[Temperature_4[x / 12]]
```

A plot of energy output as a function of time is generated.

In[70]:=
```
EnergyPlots = Show[
   Table[
    Plot[
     Energy_{solar,i}[x], {x, 0, 12},
     PlotStyle -> {Color_i}, PlotRange → {0, 500}], {i, 1, 4}],
   AxesLabel → {"Months", "Energy"}, PlotLabel → "Energy Produced by Solar Panels"];
```

In[71]:=
```
Rasterize[EnergyPlots, ImageResolution → 300]
```

Out[71]=



- **BWC Excel 10kW Class Wind Turbine:**

Start-up wind speed:

In[72]:=
```
wind_{startup} = 7.5;
```

In[73]:=
```
Wind_{speed} = Table[4.0 + 0.5 i, {i, 0, 7}] ;
```

Converting units to the desired MPH.

In[74]:=
```
Wind_{speed,MPH} = Wind_{speed} * 2.24;
```

The BWC Excel 10kW Class Turbine has the following energy production based on wind speeds.

In[75]:=
```
Power_{gen,year} = {1221, 1781, 2443, 2766, 3331, 3610, 3877, 4047};
```

Because the values are based on yearly energy production, we must divide the values by 12 in order to have them in monthly production.

In[76]:=
```
Power_{gen,month} = N[Power_{gen,year} / 12];
```

We can now couple the data into a set of data points in which we can plot and obtain a formula.

```
In[77]:= Turbine_data = Table[{Wind_speed,MPH[[i]], Power_gen,month[[i]]}, {i, 1, Length[Power_gen,month]}];
```

The data points seem to be of the Logorithmic form and therefore a fit to the generic formula c Log[x] +b.

```
In[78]:= sol_turbinedata = FindFit[Turbine_data, (c_turb * Log[x]) + b_turb, {c_turb, b_turb}, {x}];
         {c_turb, b_turb} = {c_turb, b_turb} /. sol_turbinedata;
```

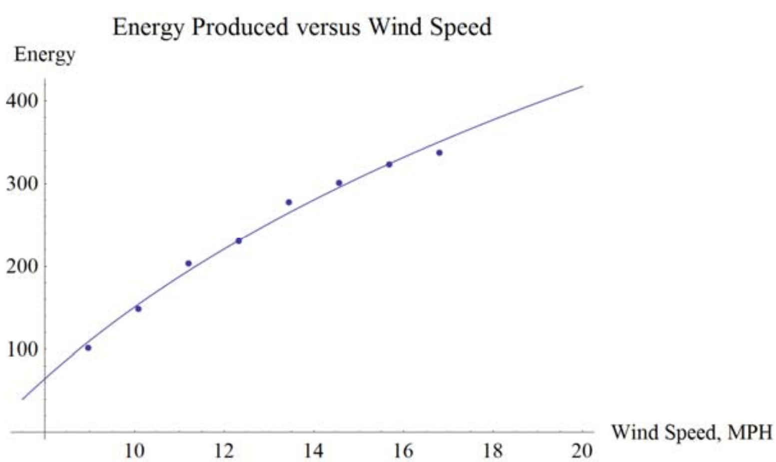
The values for the coefficients have been found and now the forumla must be defined.

```
In[80]:= Turbine_power[x_] := c_turb * Log[x] + b_turb
```

The function obtained must be checked against the data points to ensure the validity of the function.

```
In[81]:= Rasterize[Show[Plot[Turbine_power[x], {x, 7.5, 20}],
           ListPlot[Turbine_data], AxesLabel → {"Wind Speed, MPH", "Energy"},
           PlotLabel → "Energy Produced versus Wind Speed"], ImageResolution → 300]
```

Out[81]=



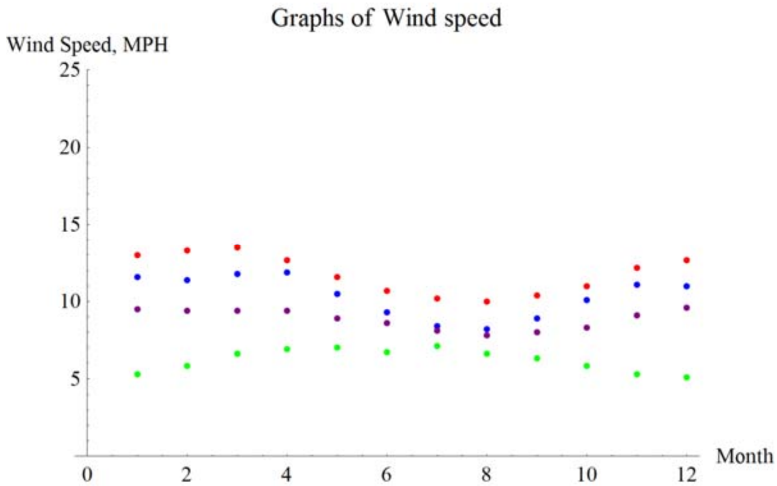
- **Obtaining formulas for the projected winds of each city:**

Plotting the average monthly wind speeds as a function of time for each city.

```
In[82]:= WindDataPlot =
           ListPlot[Table[Winddata_i = Table[{j, Wind_i[[j]]}, {j, 1, Length[Wind_i]}], {i, 1, 4}],
            PlotStyle -> {Red, Green, Blue, Purple}, PlotRange → {0, 25},
            AxesLabel → {"Month", "Wind Speed, MPH"}, PlotLabel → "Graphs of Wind speed"];
```

In[83]:=
```
Rasterize[WindDataPlot, ImageResolution → 300]
```

Out[83]=



Once again, the general form of the sin function will be used to fit a line to the data points.

In[84]:=
```
Table[
  Windsol_i = FindFit[Winddata_i, A_wind,i + B_wind,i Sin[C_wind,i (x - D_wind,i)],
    {A_wind,i, B_wind,i, C_wind,i, D_wind,i}, {x}], {i, 1, 4}];
Table[{A_wind,i, B_wind,i, C_wind,i, D_wind,i} = {A_wind,i, B_wind,i, C_wind,i, D_wind,i} /. Windsol_i, {i, 1, 4}];
```

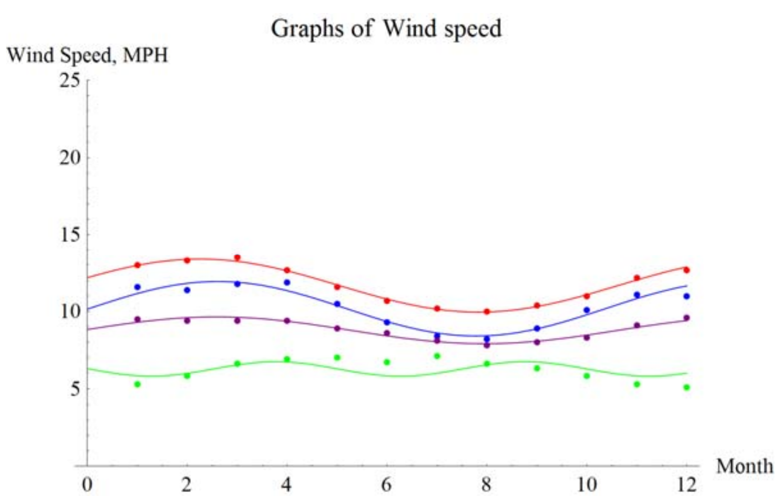
The coefficients for each function has been obtained and the functions of Windspeed for each city can now be defined.

In[86]:=
```
WindFunction_1[x_] := A_wind,1 + B_wind,1 Sin[C_wind,1 (x - D_wind,1)];
WindFunction_2[x_] := A_wind,2 + B_wind,2 Sin[C_wind,2 (x - D_wind,2)]
WindFunction_3[x_] := A_wind,3 + B_wind,3 Sin[C_wind,3 (x - D_wind,3)];
WindFunction_4[x_] := A_wind,4 + B_wind,4 Sin[C_wind,4 (x - D_wind,4)];
```

Plotting the fucntions along with the data points to ensure the vailidity of the function.

In[90]:=
```
Rasterize[Show[WindDataPlot, Plot[WindFunction_1[x], {x, 0, 12}, PlotStyle → {Color_1}],
  Plot[WindFunction_2[x], {x, 0, 12}, PlotStyle → {Color_2}],
  Plot[WindFunction_3[x], {x, 0, 12}, PlotStyle → {Color_3}],
  Plot[WindFunction_4[x], {x, 0, 12}, PlotStyle → {Color_4}]], ImageResolution → 300]
```

Out[90]=



Note: The function for the wind speed of Phoenix did not fit as nicely as the other functions, but fortunately the average wind speed in each month is below 7.5 MPH which is the necessary speed required for the start up of the turbine. And because it does not meet this requirement, Phoenix can not produce energy via wind turbines.

The data can now be combined in order to obtain formulas for the energy produced as a function of wind speed for a given city.

$N_{turbines}$ is equal to the number of turbines a city has, the default value will be 1 in order to understand how each region compare.

The function Turbine_{power} is a function of wind speed which is denoted by WindFunction. These functions are time dependant.

In[91]:=
```
N_turbines = 1
```

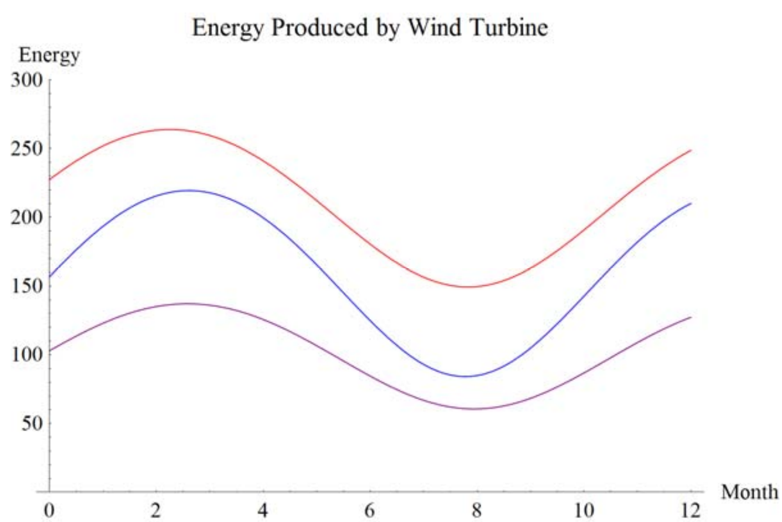
Out[91]= 1

In[92]:=
```
Energy_wind,1[x_] := N_turbines * Turbine_power[WindFunction_1[x]]
Energy_wind,2[x_] := 0
Energy_wind,3[x_] := N_turbines * Turbine_power[WindFunction_3[x]]
Energy_wind,4[x_] := N_turbines * Turbine_power[WindFunction_4[x]]
```

In[96]:=
```
WindEnergyPlots = Show[
   Table[
    Plot[
     Energy_wind,i[x], {x, 0, 12},
     PlotStyle -> {Color_i}, PlotRange → {0, 300}], {i, 1, 4}],
   AxesLabel → {"Month", "Energy"}, PlotLabel → "Energy Produced by Wind Turbine"];
```

In[97]:=
```
Rasterize[WindEnergyPlots, ImageResolution → 300]
```

Out[97]=



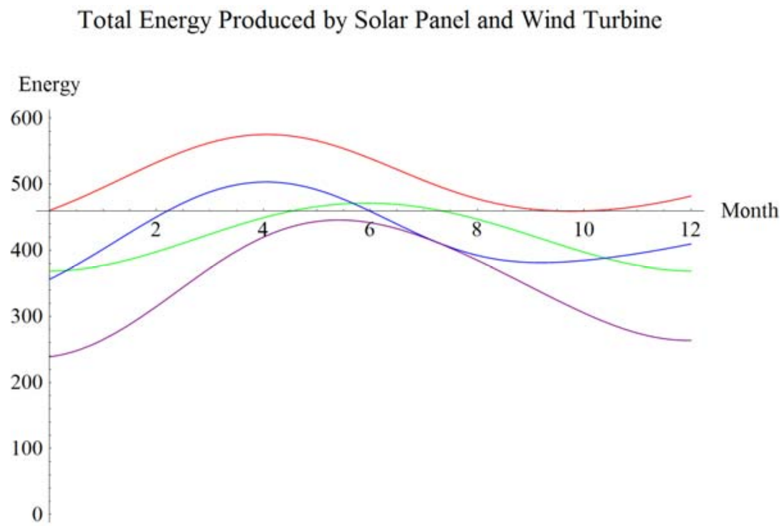
- **Combining the solar panel function with the wind turbine function:**

In[98]:=
```
Energy_total,1[x_] := Energy_solar,1[x] + Energy_wind,1[x]
Energy_total,2[x_] := Energy_solar,2[x] + Energy_wind,2[x]
Energy_total,3[x_] := Energy_solar,3[x] + Energy_wind,3[x]
Energy_total,4[x_] := Energy_solar,4[x] + Energy_wind,4[x]
```

In[102]:=
```
EnergyPlots = Show[
   Table[
    Plot[
     Energy_{total,i}[x], {x, 0, 12},
     PlotStyle -> {Color_i}], {i, 1, 4}]];
```

In[103]:=
```
Rasterize[Show[EnergyPlots, PlotRange → {0, 600}, AxesLabel → {"Month", "Energy"}, PlotLabel →
   "Total Energy Produced by Solar Panel and Wind Turbine"], ImageResolution → 300]
```

Out[103]=



■ Calculating the area under the curve.

The area under the curve is used in order to obtain the total yearly power production. The Integrate command of *Mathematica* was not working properly, so the Monte Carlo method of finding area is used.

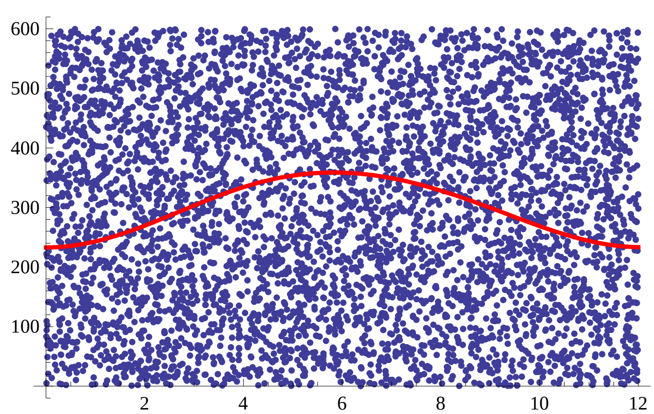
In[104]:=
```
dots = 5000;
```

In[105]:=
```
RandomPoints = Table[{x_i = RandomReal[{0, 12}], y_i = RandomReal[{0, 600}]}, {i, 1, dots}];
```

In[106]:=
```
RandomPointsPlot = ListPlot[RandomPoints];
```

In[107]:=
```
Show[RandomPointsPlot, Plot[
  Energy_{solar,1}[x], {x, 0, 12}, PlotStyle -> {Thick, Red}]]
```

Out[107]=



We can calculate the areas under the curve of the solar panels to see how much energy we are producing in each city per year.

In[108]:=
```
SolarArea = Table[Total[Table[{
        If[y_i < Energy_solar,j[x_i], 1, 0]}, {i, 1, dots}]], {j, 1, 4}
    ];
```

We can calculate the areas under the curve of the wind turbines to see how much energy we are producing in each city per year.

In[109]:=
```
WindArea = Table[Total[Table[{
        If[y_i < Energy_wind,j[x_i], 1, 0]}, {i, 1, dots}]], {j, 1, 4}
    ];
```

Total Yearly Solar Energy Saved:

In[110]:=
```
TYSES = N[(12 * 600) * (SolarArea / dots)];
```

Total Yearly Wind Energy Saved:

In[111]:=
```
TYWES = N[(12 * 600) * (WindArea / dots)];
```

Cost of kWh for each state, [cents/ kWh]:

In[112]:=
```
Cost = {15.22, 8.47, 7.91, 7.20};
```

Savings for each state:

In[113]:=
```
SolarSavings = Cost * TYSES / 100;
```

In[114]:=
```
WindSavings = Cost * TYWES / 100;
```

The values in the table below are for 1 solar panel and 1 wind turbine, these savings are per year.

In[115]:=
```
Rasterize[TableForm[Table[{City_i, SolarSavings[[i]],
     WindSavings[[i]], SolarSavings[[i]] + WindSavings[[i]]}, {i, 1, 4}],
   TableHeadings → {None, {"City", "Solar Savings, USD", "Wind Savings, USD",
      "Combined Savings, USD"}}], ImageResolution → 300]
```

Out[115]=

City	Solar Savings, USD	Wind Savings, USD	Combined Savings, USD
NEW YORK	543.975	391.215	935.19
PHOENIX	428.961	0.	428.961
CHICAGO	257.081	150.923	408.004
SEATTLE	217.417	86.5728	303.99

Lets see how much a company in NYC can save with a range of solar panels and a range of wind turbines. The maximum amout of solar panels are 12 and the maximum amount of wind turbines are 10.

In[116]:=
```
Rasterize[TableForm[Table[
    Table[{i * SolarSavings[[1]] + j * WindSavings[[1]]}, {i, 1, 12}], {j, 1, 10}],
   TableHeadings → {{"1", "2", "3", "4", "5", "6", "7", "8", "9", "10"},
     {"1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "11", "12"}}], ImageResolution → 300]
```

Out[116]=

	1	2	3	4	5	6	7	8	9	10
1	935.19	1479.16	2023.14	2567.11	3111.09	3655.06	4199.04	4743.01	5286.99	5830.
2	1326.4	1870.38	2414.35	2958.33	3502.3	4046.28	4590.25	5134.23	5678.2	6222.
3	1717.62	2261.59	2805.57	3349.54	3893.52	4437.49	4981.47	5525.44	6069.42	6613.
4	2108.83	2652.81	3196.78	3740.76	4284.73	4828.71	5372.68	5916.66	6460.63	7004.
5	2500.05	3044.02	3588.	4131.97	4675.95	5219.92	5763.9	6307.87	6851.85	7395.
6	2891.26	3435.24	3979.21	4523.19	5067.16	5611.14	6155.11	6699.09	7243.06	7787.
7	3282.48	3826.45	4370.43	4914.4	5458.38	6002.35	6546.33	7090.3	7634.28	8178.
8	3673.69	4217.67	4761.64	5305.62	5849.59	6393.57	6937.54	7481.52	8025.49	8569.
9	4064.91	4608.88	5152.86	5696.83	6240.81	6784.78	7328.76	7872.73	8416.71	8960.
10	4456.12	5000.1	5544.07	6088.05	6632.02	7176.	7719.97	8263.95	8807.92	9351.

From the chart one may notice how a company in nyc can save a maximum of roughtly $10,500 per year.