FINAL REPORT

A Framework for Estimating Emissions of Freight Transportation Operations

May 13, 2013

PI: Pablo L. Durango-Cohen, Associate Professor

Students: Yikai Chen, Madison Fitzpatrick, and Yidan Luo

Department of Civil and Environmental Engineering

A Framework for Estimating Emissions of Freight Transportation Operations

1. Executive summary

In recent years, corporations have shown increasing interest in measuring their environmental impacts, especially pollutant emissions. Business interests—for instance, preparing for imminent regulation—motivate this trend as much as ethics. Investors and customers factor environmental impact into their business decisions, and sustainable companies have a competitive advantage.

According to Department of Transportation Climate Change Report, transportation greenhouse gas (GHG) emissions have been growing steadily in recent decades. From 1990 to 2006 alone, transportation GHG emissions increased 27 percent, accounting for almost one-half of the increase in total US GHG emissions for the period. Transportation GHG emissions account for 29 percent of total US GHG emissions, and over 5 percent of global GHG emissions. In 2006, emissions from on-road vehicles accounted for 79 percent of transportation GHG emissions. Emissions from trucks accounted for 19 percent.

For companies with large distribution systems, emissions from transport operations constitute a significant portion of their environmental impact. Many models have been developed to estimate vehicle emissions, though the focus in research and in practice has been on automobiles, as opposed to trucks and other heavy vehicles. In addition, there is a lack of standards governing emissions reporting. Wide variations and lack of documentation make comparisons between companies or years difficult.

The goal of this project is, therefore, to develop a rigorous, flexible, and practical framework for estimating the emissions of freight transport operations. To meet this goal, we developed a software tool based on simple, yet defensible and reliable economic models of marginal emissions contributions from individual shipments. The initial focus is on trucking, but extensions for other freight modes are possible.

The tool provides several practical advantages over currently available methods. In particular, it:

- Is based on data that are commonly available from freight carriers;

- Uses models of marginal emissions contributions instead of aggregate models;

- Performs detailed analysis, by route, carrier, and business sector;

- Provides error bounds, describing the precision of emissions estimates; and

- Provides quantitative support for strategic shipping and supply-chain decisions.

The tool is intended to be deployed as an online tool, and consulting services could be offered to users in order to support their use of the tool for emissions management.

2. Background

Large companies are becoming increasingly interested in measuring and improving the sustainability of their operations. Driving this trend are several emerging factors that give sustainable companies an advantage over their less sustainable competitors. Investors, customers, and other participants in economic markets are concerned about damage to the environment cased by economic activity, and some are using their power to persuade manufacturers and distributors to decrease their environmental impact.

For example, the Global Responsible Investment Network (GRIN), formed in 2002, publishes an annual list of the 100 most sustainable corporations in the world. The mission of GRIN is to create a business environment in which the most sustainable companies attract the most capital and earn greater returns.

In US, the EPA encourages corporate sustainability through its voluntary SmartWay program. In addition to a positive reflection on their reputations, companies affiliated with SmartWay receive benefits, including branding, access to financing, and lower fuel costs. This creates an incentive structure that makes it beneficial for carriers and shippers to track and reduce their environmental impact.

In Europe, the transport and environment reporting mechanism (TERM) was set up by the European Environment Agency (EEA) on request of the EU transport ministers in 1998. The main aim of TERM is to monitor the progress and effectiveness of transport and environment integration strategies on the basis of a core set of indicators. TERM helps to track the environmental performance of transport in the EEA member countries. EEA also works on transport and environment issues including transport emission inventories, air quality work, climate change and transport noise monitoring. In addition, the European Commission supported a large research project, ARTEMIS, with 40 European laboratories over 5 years, to improve the European methods for estimating and inventorying the pollutant emissions from the transports. It concerned all transports modes in Europe.

Many companies are also preparing for the possibility of government regulation of corporate emissions. This could potentially come in the form of carbon taxes or an emissions cap-and-trade system. In 2009, President Obama proposed a cap-and-trade plan with ambitious emissions reduction goals. Already planned or in effect in the US are several programs that regulate emissions in some manner. In addition, a number of legislative proposals for carbon taxes or some form of cap-and-trade have been submitted in the 111th Congress.

The aforementioned trends, programs and policies rely on the ability to measure or estimate emissions, which is not simple. Many models, spanning a large range of complexity, have been developed to estimate vehicle emissions. However, there are currently no standards to guide researchers, corporations, and other interested parties in the application of these models. Moreover, the focus of most models has been on automobiles, as opposed to trucks or other heavy vehicles.

In addition, there seems to be little understanding or recognition in the corporate

world of the limitations of these models. No estimation model is absolutely accurate, but most corporate sustainability reports include no discussion of the accuracy or precision of their numerical emissions estimates. Also, methodological differences, often not documented in corporate reports, make comparisons between companies or years difficult or invalid.

As discussed above, there is a strong potential for future regulation of corporate emissions. The project, based on the most accurate and cost-effective estimation methodologies available, is intended to result in a number of benefits to corporations seeking to measure and improve their emissions. Users of the tool would gain an understanding of the accuracy of their emissions estimates. Guidelines and standards for data collection and estimation procedures would lead to improved accuracy in tracking emissions over time. The use of disaggregate modeling would also give companies the ability to identify target areas for improvement in their transport policies. If a large enough number of companies use this tool, researchers and industry analysts may be able to use their data to identify new best practices in transportation and distribution logistics.

3. Literature survey

<u>Academic Literature</u>

Most of the vehicle emission models that have been published in the academic literature are of limited practical use. The models tend to be very complex and require input data that are not widely available in the commercial surface transportation sector today, such as instantaneous speed and acceleration, or traffic flow characteristics. Although some modern, highly equipped trucks can measure these variables, carriers typically do not record this data because storing and managing it would be a major challenge. However, it is useful exercise to understand how far the academic study of emissions modeling has come, and so the five major categories of models are summarized below.

- *Average speed.* Some models calculate emissions based on mean traveling speed and distance traveled. Vehicle emissions are highly dependent on speed, but emissions based on average speed may differ greatly from actual emissions if the speed over the journey varies widely.

- *Traffic situation.* Emissions can also be estimated based on characteristics of the traffic in which a vehicle is traveling. These models track the distance traveled in each of a number of discrete situations, such as free-flow highway, uncongested urban arterial, stop-and-go traffic, or others.

- *Traffic variables.* The traffic situation concept can be carried a step further by explicitly accounting for quantitative characteristics of the traffic flow such as average speed, density, queue length, and signal settings.

- *Driving mode.* One can also estimate emissions as a function of engine or vehicle operating mode. This is calculated from the same inputs used in driving variable models but at the highest resolution.

A practical disadvantage of the above described models is that they are based the movement of vehicles, not the movement of shipments. This makes it difficult to define the marginal impact of a shipment.

<u>Corporate transportation emissions</u>

The academic models described are not likely to be useful measuring the transportation emissions of large companies. These models require large amounts of high-quality data, which in most cases are not available. In our experience with one large corporation, the input data for emissions estimation have come from carriers and freight payment providers. In the best case, there is information about the distance traveled, weight and mode of each shipment. Sometimes even these details are not available, or they are not exact. The same is likely to be true for other companies seeking to estimate transport emissions. So the focus in this project is on simpler models that are feasible given currently available data.

<u>Practical use</u>

SmartWay is a widely used EPA program that calculates transportation-related

emissions. It uses a very simple, aggregate method for estimating emissions based on national average values for fuel economy and the quantity of emissions produced per gallon of fuel burned. In conjunction with an estimate of distance driven, one can use this information to quickly estimate total emissions produce during a trip. The general equation for calculating CO_2 emissions using fuel consumption values is

$$CO_2 = Fuel_i \times EF_i$$

$Fuel_i$ = volume of fuel combusted of type i

EF_i = Fuel Emissions Factor of fuel type i

This is a fuel-based approach. The emission factor depends on the type of fuel, type of truck and whether the truck is fully loaded or empty. Sometimes, when the fuel quantity is not directly available, this approach will collect data on distance traveled by vehicle type and fuel type and then convert distance traveled data into fuel use values based on fuel economy factors. The general equation for the conversion is

$$Fuel = D \times FEF_i$$

D = Distance traveled

FEF_i = Fuel Economy Factor of fuel type i

It uses previously aggregated data at fleet level, not each shipment, which means average value or total value of a fleet. To be more specified, the input data SmartWay needs for calculation include the following.

- Total fuel used or total distance
- Type of fuel
- Average payload
- Average capacity volume
- Types of trucks and each share in the fleet
- Road type

SmartWay estimates fuel emissions factor for each fleet and then estimate the total emissions, only based on the average level. This approach needs a relatively small amount of data that are easy to get, and it does not require information for each shipment. The final output of SmartWay contains a variety of parameters of the fleet, such as the following.

- Miles per gallon
- Miles per truck
- Tons of CO_2
- Grams CO_2 per mile
- Grams CO_2 per ton-mile

SmartWay can also make comparison across years for companies. The year-to-year comparison report allows the user to compare the fleet/division characteristics and activity values, as well as CO_2 performance metrics across two different reporting years. Percentage changes are also indicated for each of the output items mentioned above, relative to the earlier year baseline values.

The obvious disadvantage of this method is that the recommended values for fuel economy are averages based on assumptions about fuel quality and oxidation rate. Studies have shown that production of different types of emissions is not necessarily related linearly to the amount of fuel burned. However, this method uses the type of data are available to us, and the assumptions and values can be adjusted for the specific fleet. Another disadvantage is that the analysis it produces is very general. There is no specific analysis based on different business sector, route or carrier, although this information is often available in shippers.

COPERT (COmputer Programme to calculate Emissions from Road Transport) is a macro level emission estimation tool for road transport. It was developed by the EEA and is used by many European countries for reporting official emissions data. It includes a methodology to estimate vehicle fleet emissions on a country level, trying to balance the need for detailed emission calculations with the desire to minimize the amount of input data required. It also uses emission factors as the core of the estimation model. The model is primarily used for policy and decision analysis on nation level. For example, it has been used in impact assessment studies of the European commission to evaluate the impact of proposed technological and legislative measures to road transport. Also this approach is based on national or regional average values of factors and the data it uses are simple and aggregate, which is not applicable for company level. It may be helpful for a nation's decision and policy makers, but it has little applicability for shippers and carriers.

A New Solution

This project attempts to create a rigorous, flexible and practical approach to estimate emissions from freight transport operations. Existing methods are simplistic and aggregate, based on average rather than marginal contributions to emissions. Our approach is based on formulating simple, yet defensible and reliable economic models of marginal emissions contributions from individual shipments.

Our approach builds on data that are already commonly collected by freight companies. This contrasts with many emissions estimation models in the academic literature, which rely on input data that are very rarely available, such as instantaneous speed, acceleration, and grade. Freight carriers do record information about a shipment's origin and destination, weight, volume, and mode. Additional data may include further detail about the type of shipment or factors affecting the fuel efficiency of the shipment. The data category and basic equations we use are the same as those used in SmartWay, but we make use of data at the shipment level, when they are available, rather than aggregate level. Calculating marginal emission provides users with error bounds on emissions estimates and quantitative support

for strategic decisions. Another practical advantage of this approach is that the information is modular. If a particular piece of information is not available, it can be replaced with an assumption. For example, if details about the fleet are not available, national average values can be substituted. This allows for quantitative analysis of the value of additional data in terms of estimation accuracy.

In addition, calculating emissions at shipment level allows for more detailed analysis. The total emissions and average emissions can be summarized for each route, carrier and business sector. As a result, the model is well suited to support decision-making by quantifying the impact of various shipping decisions.

Our approach makes better use of data at a disaggregate level than SmartWay, and it can analyze accuracy, so our emissions estimates will be more informative than those provided by other online tools.

4. Technical description of the approach

This section provides detailed background information on the estimation methods, input data and assumptions used within our approach. The main purpose is to develop a tool that is capable of performing detailed analyses of freight truck emissions at the level of individual routes, product divisions, and carriers. Our approach calculates emissions by multiplying shipment data with EPA-developed emissions rate factors.

Before calculating CO_2 emissions, the first piece of information that needs to be determined is the quantity of fuel combusted, so the best way to calculate CO_2 emissions is by fuel combustion. But there are two approaches for determining how much fuel has been used: top-down and bottom-up. The most accurate method of determining the amount of fuel combusted, and therefore the preferred method, is a top-down approach, which accounts for the total amount of fuel used in mobile sources. In the top-down approach, the firm knows how much fuel was consumed on each shipment from fuel receipts or direct measurement in the truck. In the bottom-up approach, fuel use is calculated based on vehicle activity, which is usually characterized in terms of miles traveled, weight, and speed distribution.

Although the top-down approach is more accurate, many trucking firms (especially small ones) do not keep good records of fuel use, and they often do not share this information with the shippers that hire them. Because we are designing this tool for carriers and shippers, we should not expect them to have accurate top-down estimates of fuel use. However, we should give them the option of using such data if they are available.

The tool gives users three options so that they can select the best calculation method based on the data they have. The first option is based on top-down approach, which is the most accurate estimate and requires fuel use data. As fuel efficiency is the most important factor to decide fuel use, if the fuel use data are not available, the tool provides other two options. Some shippers have information on the fuel efficiency, which can give a more accurate estimate of fuel use and emissions. So the second option is based on bottom-up approach with fuel efficiency for each shipment known. The third option does not require any information about fuel use and fuel efficiency. The user only has to provide distance traveled, weight and fuel type for each shipment.

Once the amount of fuel combusted is determined, the next step in calculating emissions is to determine how much carbon is in the fuel. This can be determined from fuel density and carbon fraction directly, or by heat content and carbon content per unit of energy. The three different options in our tool have the same equation and methodology in this step. We assume emission factors are independent of the truck type, speed and other operating practices. The methodology and relative emission factors are based on EPA GHG inventory protocol, as shown in the following content.

CO_2 emissions are calculated for each shipment based on the data input option

chosen by the user. The tool contains slightly different equations when determining the fuel use and emissions for each option.

For option 1, the emission equation used for each shipment is

$$CO_2 = Fuel_i \times CC_i \times FO_i \times \frac{CO_{2(m.w.)}}{C_{(m.w.)}}$$

$Fuel_i$ = volume of fuel combusted of type i

CC_i = carbon content per unit volume of fuel type i

FO_i = fraction oxidized of fuel type i

CO_2 (m.w.) = molecular weight of carbon dioxide

C (m.w.) = molecular weight of carbon

For option 2, the emission equation used for each shipment is

$$CO_2 = \frac{d}{FE_i} \times CC_i \times FO_i \times \frac{CO_{2(m.w.)}}{C_{(m.w.)}}$$

d = distance traveled

FE_i = fuel economy using fuel type i

CC_i = carbon content per unit volume of fuel type i

FO_i = fraction oxidized of fuel type i

CO_2 (m.w.) = molecular weight of carbon dioxide

C (m.w.) = molecular weight of carbon

For option 3, the emission equation used for each shipment is

$$CO_2 = d \times w \times \frac{EE_i}{HC_i} \times CC_i \times FO_i \times \frac{CO_{2(m.w.)}}{C_{(m.w.)}}$$

d = distance traveled

w = weight

EE_i = default energy economy of heavy-duty trucks

HC_i = heat content per unit volume of fuel type i

CC_i = carbon content per unit volume of fuel type i

FO_i = fraction oxidized of fuel type i

CO_2 (m.w.) = molecular weight of carbon dioxide

C (m.w.) = molecular weight of carbon

Table 1: Factors for diesel and gasoline[1]

	Diesel	Gasoline
CC: Carbon Content (kg C / gal)	2.77	2.40
FO: Fraction Oxidized	1.00	1.00
$CO_{2\,(m.w.)} / C_{(m.w.)}$ ratio	44/12	44/12
EE: Default Energy Efficiency (BTU/ton-mi)	3,200	3,200
HC: Heat Content (BTU/gallon)	139,200	125,000

[1]EPA Office of Air and Radiation. 2008. *Climate Leaders Greenhouse Gas Inventory Protocol Core Module Guidance: Direct Emissions from Mobile Combustion Sources.* www.epa.gov/climateleaders.

As stated above, the tool estimates the emissions based on known emission factors and activity data. The following list summarizes the input data required to run the model.

- Shipment ID (required)

- Origin and destination (required)

- Fuel type (required)

- Carrier (optional)

- Business sector (optional)

- Mode (optional)

- Weight in pounds (required for option 3)

- Distance in miles (required for option 3)

- Quantity in gallons and efficiency of fuel used in miles per gallon (required for option1 and 2, respectively)

Output includes total and average (per shipment) carbon dioxide emissions. As the emissions are calculated for each shipment, this marginal contribution can be summarized by origin-destination pair, business sector, carrier, or any combination of these, as long as the user provides sufficient information.

The user can select analysis option according to the need. For example, if the user chooses to analyze emissions by carrier and route, the tool outputs a list of all the carriers, and within each carrier, each route that the carrier serves. For each item, the program provides total emissions and average emissions per shipment on that carrier and route.

5. Empirical comparison

This section is designed to empirically show that our tool is more flexible, precise, and rigorous than the existing SmartWay tool. A Fortune 500 company has established a relationship with Northwestern University Transportation Center and has contracted with NU for the past three years to develop the North American Trucking section of their annual sustainability report. This company provided data for this empirical test. The following tables show the result from SmartWay and our tool.

Table 2: result of SmartWay

Fuel used	74,873 gallons
Distance	159,837 miles
CO2 emissions	733 tonnes 1,630 grams per Mlb

Table 3: result of our tool

Total Emissions (kg)	Total Shipments	Average Emissions (kg/shipment)		
733,733	29	25,301.14		
Analysis by Business Sector				
Business Sector	Total Emissions (kg)	Total Shipments	Average Emissions (kg/shipment)	
DOM APPL & PERS CARE	626623	8	78327.88	
PHILIPS ELECTRONICS	107109.9	21	5100.473	
Analysis by Route				
Origin	Destination	Total Emissions (kg)	Total Shipments	Average Emissions (kg/shipment)
BOTHELL	ROSELLE	1046.285	1	1046.285
GROVEPORT	ANDERSON	50.8754	1	50.8754
GROVEPORT	BETHLEHEM	93.02376	1	93.02376
GROVEPORT	BRAMPTON	641.5111	1	641.5111
GROVEPORT	BROCKPORT	431.3508	1	431.3508
GROVEPORT	BUFFALO	816.3897	1	816.3897
GROVEPORT	FLAGSTAFF	140.5252	1	140.5252
GROVEPORT	JUPITER	576.7419	1	576.7419
GROVEPORT	KUTZTOWN	137.3027	1	137.3027

GROVEPORT	ROSELLE	1213.961	1	1213.961
MORENO VALLEY	BUCKEYE	104.0267	1	104.0267
MORENO VALLEY	BUFFALO	2927.859	1	2927.859
MORENO VALLEY	CHEHALIS	5301.049	1	5301.049
MORENO VALLEY	COLDWATER	25.18305	1	25.18305
MORENO VALLEY	COLLEGE PARK	3481.641	1	3481.641
MORENO VALLEY	DALLAS	36877.71	1	36877.71
MORENO VALLEY	DE SOTO	1112.095	1	1112.095
MORENO VALLEY	DESOTO	640.7018	1	640.7018
MORENO VALLEY	EAST RANCHO DOMING	128.01	1	128.01
MORENO VALLEY	ELIZABETH	13847.12	1	13847.12
MORENO VALLEY	GROVEPORT	38848.75	1	38848.75
MORENO VALLEY	HOPE MILLS	14.2247	1	14.2247
MORENO VALLEY	ROSELLE	6490.021	1	6490.021
PHILADELPHIA	MORENO VALLEY	913.843	1	913.843
PHILADELPHIA	ROSELLE	335.8626	1	335.8626
ROSELLE	BRANTFORD	1210.009	1	1210.009
ROSELLE	MARKHAM	6854.589	1	6854.589
ROSELLE	OTTAWA	107.1719	1	107.1719
ROSELLE	WOODLAND	609365.1	1	609365.1

Since there is no fuel use or fuel efficiency in the data set, the methods both tools use to estimate carbon emissions are the same. The total emissions are the same.

It is easy to see that our tool provides a more detailed analysis. The analysis of route and business sector in this example seems trivial, but for large datasets, it can help companies see in detail the level of emissions that each part of their business is responsible for.

6.   Conclusion

This tool is based on the most accurate and cost-efficient estimation methodologies available. It is rigorous and flexible, and it can provide detailed analyses of emissions, which can support strategic decision-making. Unlike existing tools, it allows the user to explore the data at different levels of aggregation, which helps inform managerial decisions aimed at improving sustainability. Users will be able to download the tool from the web, and those who are looking for additional aid in interpreting and using the results of their analysis would benefit from consulting services from the developers.

There are several possible directions for future work on this tool. There is the possibility for extension to other emissions such as nitrous oxides (NOx) and particulate matter (PM). Carbon dioxide production is strongly related to distance and weight of a shipment and not very dependent on specific vehicle types. In contrast, NOx and PM are very dependent on fuel type, engine model year, pollution control technology installed in the vehicle, the speed distribution over the course of the trip, and other factors. SmartWay has function for calculation of NOx and PM because they have the required data. It may be necessary to directly contact the SmartWay developers to establish a data/knowledge-sharing agreement in order to access those data. Another possible extension is to include other freight modes so that companies using multiple modes to transport their products can estimate all their transport emissions using a single platform. This would be especially helpful for dealing with single shipments that travel by multiple modes.

Based on the marginal model we use in the tool, we could calculate the error bounds of the resulting emissions estimate. Because the input data to the model are modular, we can determine different bounds for data sets of varying quality. As the data improves, the estimate will become more precise, with smaller confidence intervals. We will then estimate and compare the costs and benefits of increased precision in emission estimation. It is important to develop the understanding the value of data in the estimation process. This process has not yet been fully developed.

7.   Acknowledgments

Appendix A: Implementation Platform Introduction

The tool is developed using the Visual Basic for Application platform. This section is designed to introduce the user to the tool. The instruction covers:

- Data import

- Selecting an option for calculation based on data

- Entering/submitting your data

- Outputting the result

Before using the tool, the user must save the Microsoft Excel file in a folder of your choice on your PC. To run the tool, double-click on the file to open it, and then select Enable Macros/this content Button in the security alert box. The following screen will appear.
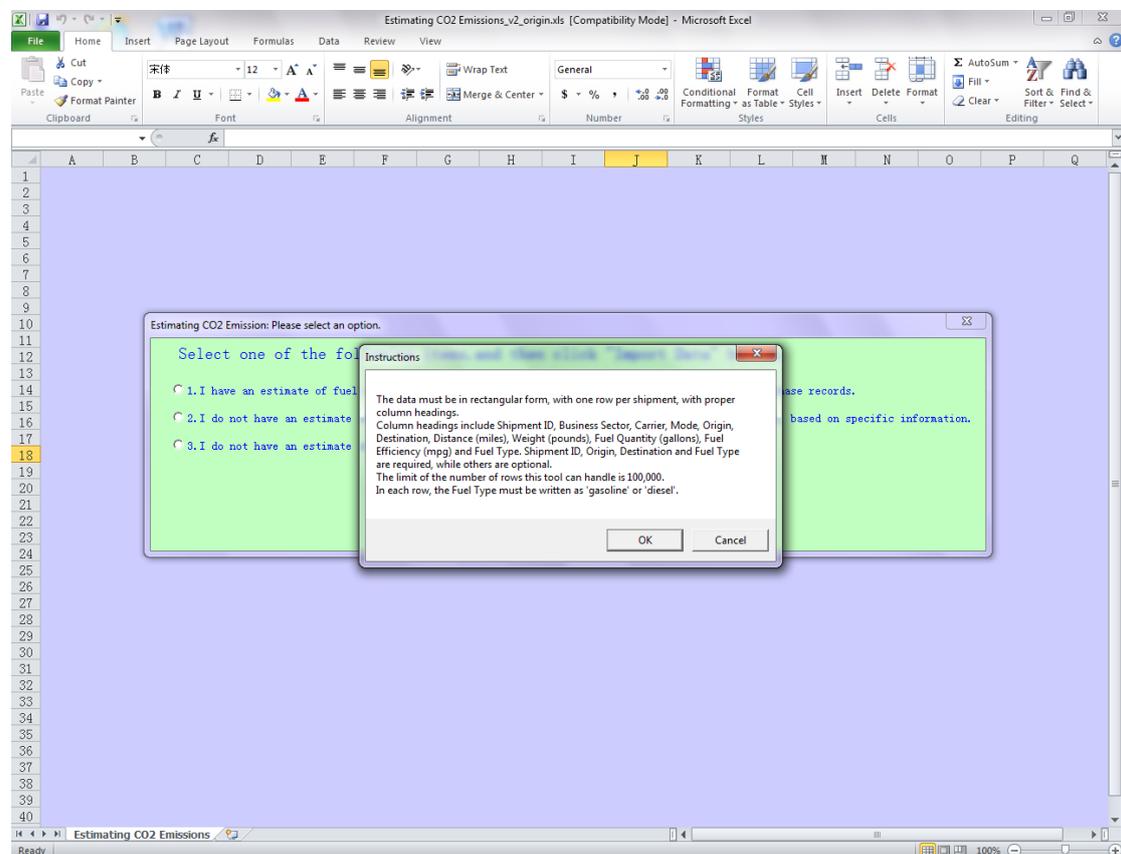
Figure 1: data instruction

This is the data format instruction so that the user can import data properly. Click "OK" button to continue.

Then the screen shown in Figure 2 should appear right after the instruction is carefully read.

The prompt asks the user to select one of the options based on the data that they have. It will help the user properly select the estimating methods and import data.

Then the user can select and open their data file, which must be an Excel document.
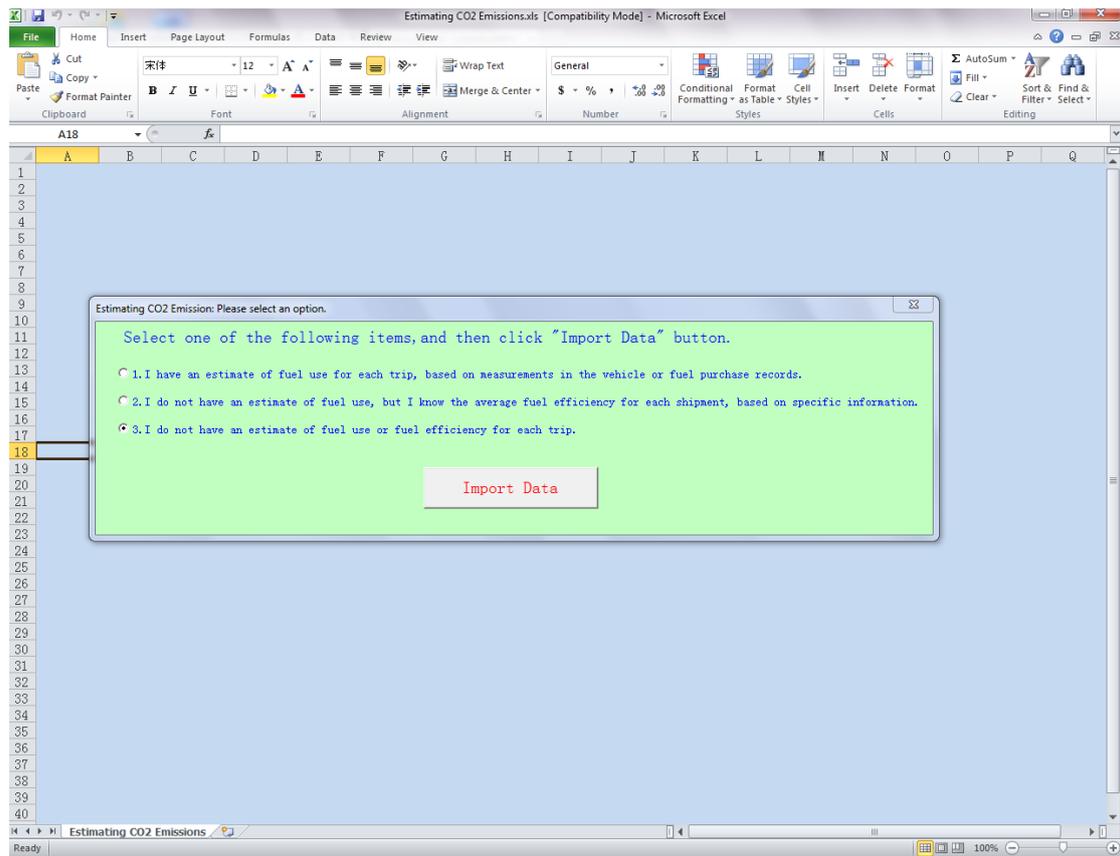
Figure 2: importing data

The tool will validate the data format before calculation.

If the required items, mentioned in section 4, are missing for selected option, the program will output an error message. The program will stop if an error message appears, so the user will have to select a different option or modify the data.

If the optional items are missing for the selected option, the program will output a warning message, so that the user knows that those items contribute to the accuracy of the emissions calculation and allow the user to perform more flexible and informative analysis. However, the program will continue to calculate carbon emissions and output analysis results.
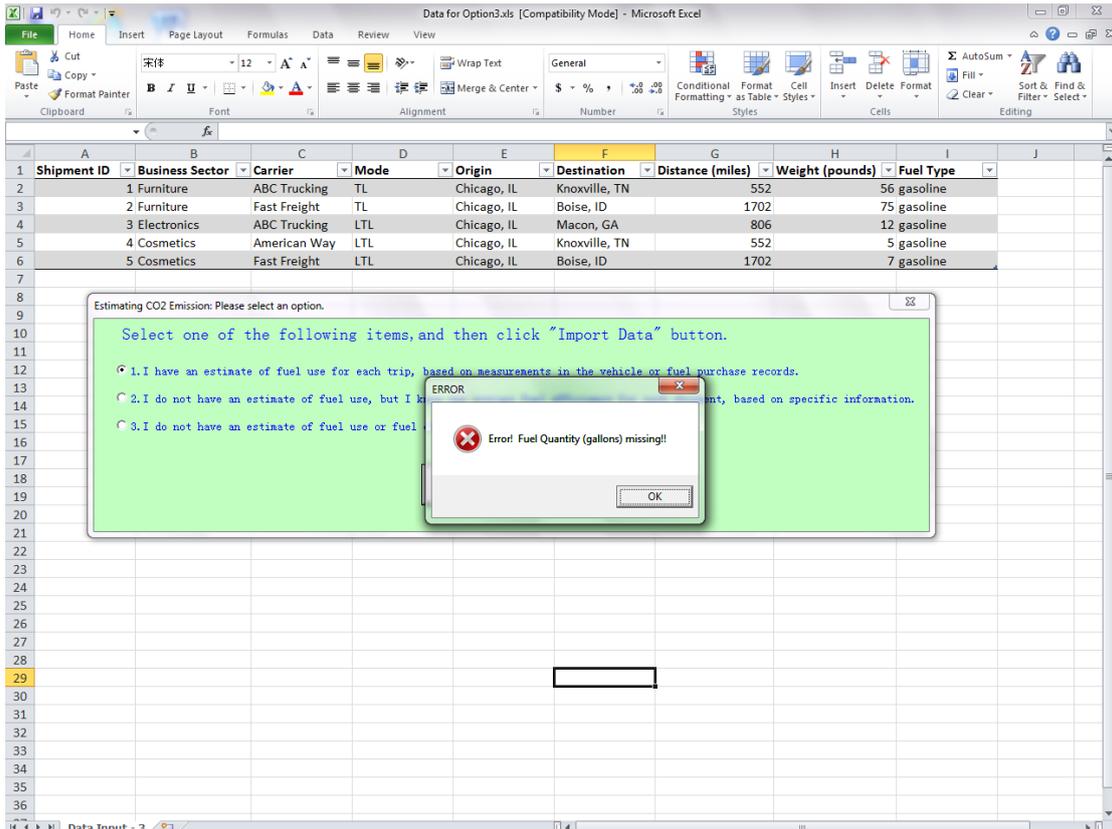
Figure 3: error message
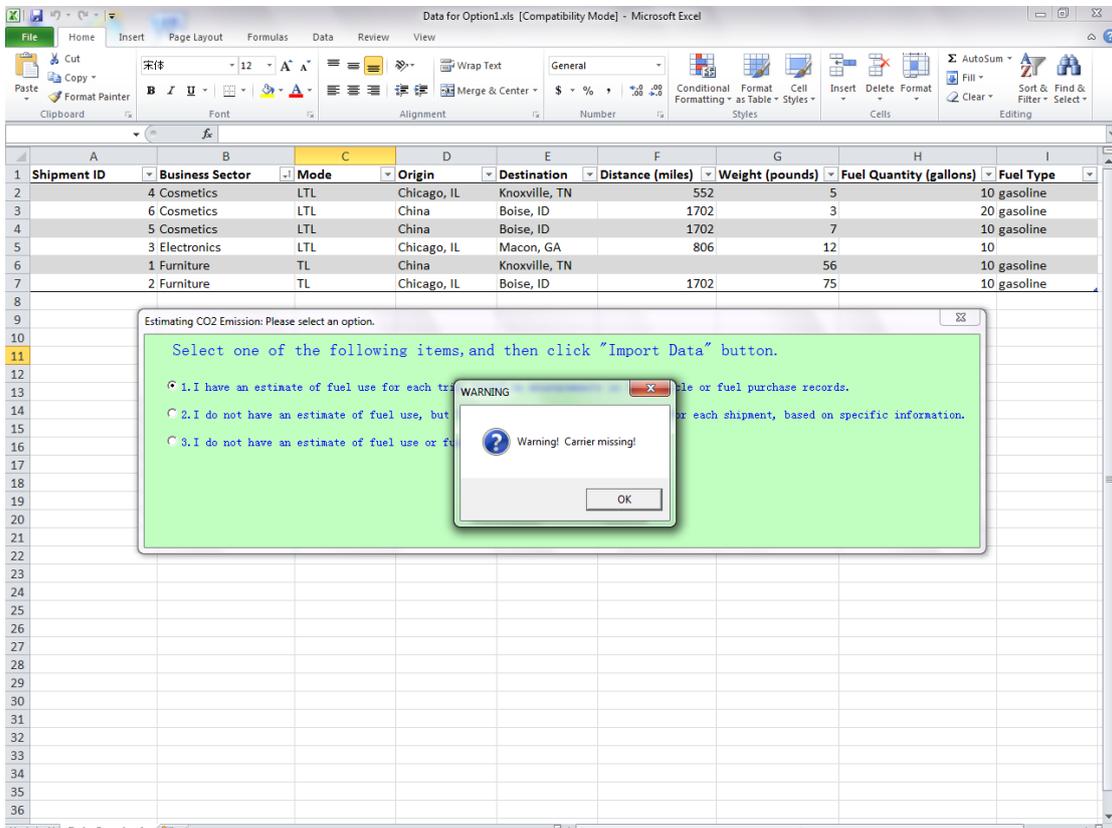
Figure 4: warning message

If the data are in the correct format and the option is correctly selected, then the tool will sum the emissions according to the user's selected analysis option. For example, if the user chooses to analyze emissions by carrier and route, the program should output a list of all the carriers, and within each carrier, each route that the carrier serves. For each item, the program should give total emissions and average emissions per shipment on that carrier and route. A screen for selecting analysis items will appear. The user can select any item that he wants to analyze. Then click the button of "select the estimating item", and the tool will output the analysis result in another Excel file.
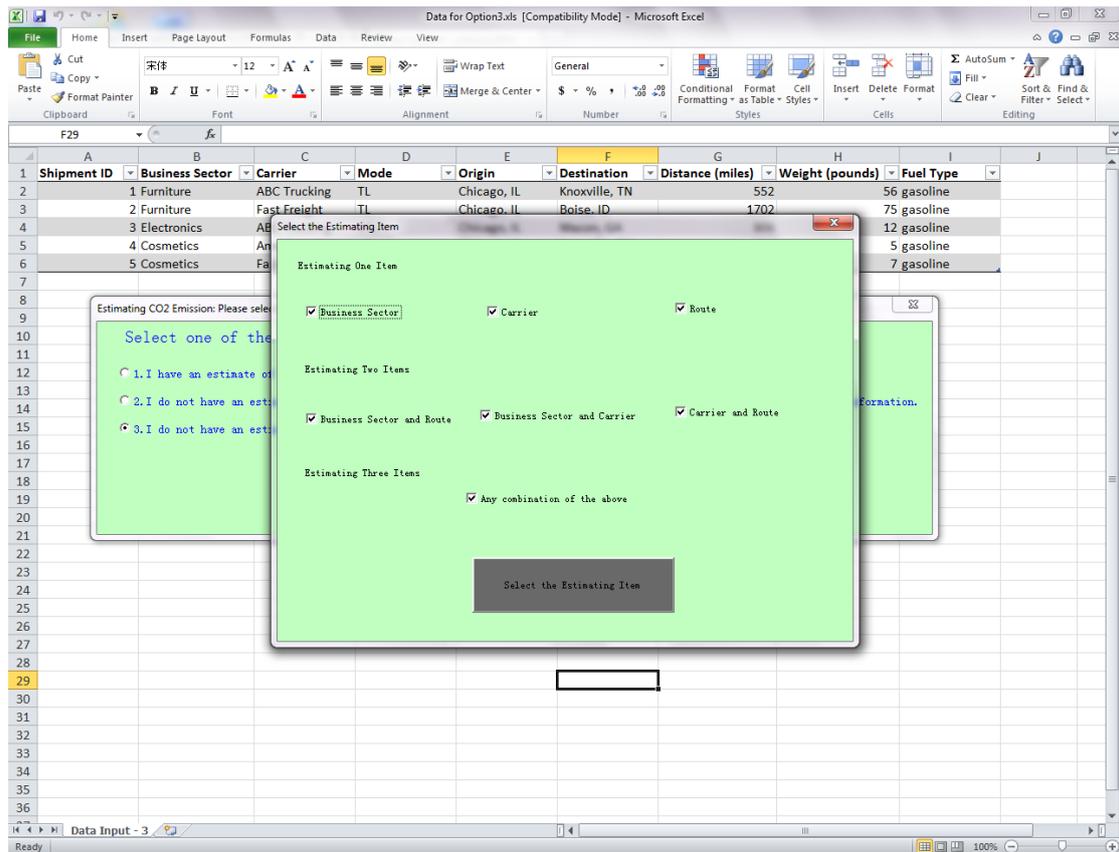
Figure 5: selecting analysis items

Before getting the result, the user will also be able to choose where to locate the output and name the file. To view the result, the user can open the saved file and all the analysis items are available.

Figure 6 gives an example of output analysis result.

The main code for the tool is located in Appendix B.

Figure 6: an example of output

Appendix B: Main Code

```vba
Dim garSourceColumnTitle()        As Variant
Dim garCountColumnTitle()         As String
Public giShowSheet                As Integer
Public garShowColumn()            As Integer
Dim gSourceColumn      As Integer
Dim gSourceRow         As Integer
Dim garCO2()           As Double
Dim gCO2               As Double
Dim gOptionType        As Integer
Dim gOutPutFile        As String
Dim gvaTitle               As Variant
Public giSector            As Integer
Public giCarrier           As Integer
Public giOrigin            As Integer
Public giDestination       As Integer


Function opt1s(iOptionType As Integer)
    Dim fdOpenSouce As FileDialog
    gOptionType = iOptionType
    Select Case gOptionType
    Case 1
        garSourceColumnTitle = Array("Shipment ID", "Business Sector", "Carrier", "Mode", "Origin", "Destination", "Distance (miles)", "Weight (pounds)", "Fuel Quantity (gallons)", "Fuel Type")
    Case 2
        garSourceColumnTitle = Array("Shipment ID", "Business Sector", "Carrier", "Mode", "Origin", "Destination", "Distance (miles)", "Weight (pounds)", "Fuel Efficiency (mpg)", "Fuel Type")
    Case 3
        garSourceColumnTitle = Array("Shipment ID", "Business Sector", "Carrier", "Mode", "Origin", "Destination", "Distance (miles)", "Weight (pounds)", "Fuel Type")
    End Select

    gOutPutFile = ThisWorkbook.Path & "\outputdata.xlsx"

    If Dir(gOutPutFile, vbNormal) <> "" Then
On Error GoTo CloseFile
        Kill gOutPutFile
    End If
On Error Resume Next
    Set fdOpenSouce = Application.FileDialog(msoFileDialogFilePicker)
    fdOpenSouce.AllowMultiSelect = True
    fdOpenSouce.Filters.Add "Excel", "*.xls,*.xlsx"
    fdOpenSouce.Title = "Please select data file"
    If fdOpenSouce.Show = True Then
```

```vba
            Dim flSelectFile As Variant
            For Each flSelectFile In fdOpenSouce.SelectedItems
                Dim wbSourceBook As Workbook
                Dim shSourceSheet As Worksheet
                Set wbSourceBook = Workbooks.Open(flSelectFile)
                Set shSourceSheet = wbSourceBook.Sheets(1)
                gSourceColumn = shSourceSheet.Cells(1, Columns.Count).End(xlToLeft).Column
                gSourceRow = shSourceSheet.Cells(Rows.Count, 1).End(xlUp).Row

                If fnCheckSourceFile(shSourceSheet) = False Then
                    MsgBox "Error! Your datasheet may lose some items that are required."
                    Exit Function
                End If

                If fnGetCalcuateSheet(shSourceSheet) = False Then
                    MsgBox "Warning! Your datasheet has some mistakes for calculation."
                End If

                Call fnCalculateCO2(shSourceSheet)
            Next
        Else
            MsgBox "Please choose the resouce file."
            Set fdOpenSouce = Nothing
            Exit Function
        End If

        wbSourceBook.Saved = True
        wbSourceBook.Close
        Set shSourceSheet = Nothing
        Set fdOpenSouce = Nothing
        MsgBox "The analysis results have been saved to the folder that you just selected."
        Exit Function
CloseFile:
        MsgBox "Please close file " & gOutPutFile, vbCritical, "ERROR"
End Function

Function fnCheckSourceFile(shSourceSheet As Worksheet) As Boolean
        Dim i                   As Integer
        Dim arMustItem()        As String
        Dim arOptionItem()      As String

        fnCheckSourceFile = False
```

```vba
ReDim arMustItem(4) As String
arMustItem(0) = "Shipment ID"
arMustItem(1) = "Origin"
arMustItem(2) = "Destination"
arMustItem(3) = "Weight (pounds)"
Select Case gOptionType
Case 1
    arMustItem(4) = "Fuel Quantity (gallons)"
Case 2
    arMustItem(4) = "Fuel Efficiency (mpg)"
Case 3
    arMustItem(4) = "Distance (miles)"
End Select

ReDim arOptionItem(4) As String
arOptionItem(0) = "Business Sector"
arOptionItem(1) = "Carrier"
arOptionItem(2) = "Mode"
arOptionItem(3) = "Distance (miles)"
arOptionItem(4) = "Fuel Type"

shSourceSheet.Activate
gvaTitle = shSourceSheet.Range(Cells(1, 1), Cells(1, gSourceColumn))

For i = LBound(arMustItem) To UBound(arMustItem)
    If Application.IsError(Application.Match(arMustItem(i), gvaTitle, 0)) Then
        MsgBox "Error!    " & arMustItem(i) & " missing!!", vbCritical, "ERROR"
        Exit Function
    End If
Next

ReDim garCountColumnTitle(2) As String
garCountColumnTitle(0) = "Shipment ID"
garCountColumnTitle(1) = "Origin"
garCountColumnTitle(2) = "Destination"

For i = LBound(arOptionItem) To UBound(arOptionItem)
    If Application.IsError(Application.Match(arOptionItem(i), gvaTitle, 0)) Then
        MsgBox "Warning!    " & arOptionItem(i) & " missing!", vbQuestion, "WARNING"
    ElseIf (arOptionItem(i) = "Business Sector") Or (arOptionItem(i) = "Carrier") Or (arOptionItem(i) = "Mode") Then
        ReDim Preserve garCountColumnTitle(UBound(garCountColumnTitle) + 1)
        garCountColumnTitle(UBound(garCountColumnTitle)) = arOptionItem(i)
    End If
```

```
        Next
        fnCheckSourceFile = True
End Function


Function fnGetCalcuateSheet(shShourceSheet As Worksheet) As Boolean

        Dim i                   As Integer
        fnGetCalcuateSheet = False

        giSector = 0
        giCarrier = 0
        giOrigin = 0
        giDestination = 0

        For i = 1 To gSourceColumn
            If gvaTitle(1, i) = "Business Sector" Then
                giSector = i
            ElseIf gvaTitle(1, i) = "Carrier" Then
                giCarrier = i
            ElseIf gvaTitle(1, i) = "Origin" Then
                giOrigin = i
            ElseIf gvaTitle(1, i) = "Destination" Then
                giDestination = i
            End If
        Next
        OptionTitleForm.Show
        fnGetCalcuateSheet = True
End Function


Function fnCalculateCO2(shSourceSheet As Worksheet) As Boolean

        Dim i                       As Integer
        Dim j                       As Integer
        Dim iCount                  As Integer
        Dim iFuelQuantityColumn     As Integer
        Dim iFuelTypeColumn         As Integer
        Dim iWeightColumn           As Integer
        Dim arResultData()          As String
        Dim strTitle                As String
        Dim strTemp()               As String
        Dim arSourceCellData        As Variant
        Dim strFuelType             As String
        Dim blCompare               As Boolean
        Dim iShowColumn             As Integer
```

```vba
    Dim wkResult As Workbook
    Dim shResult As Worksheet

    fnCalculateCO2 = True

    ReDim garCO2(gSourceRow - 1)

    arSourceCellData = shSourceSheet.Range(Cells(2, 1), Cells(gSourceRow, gSourceColumn))

    Select Case gOptionType
    Case 1
        iFuelQuantityColumn       =       shSourceSheet.Range(Cells(1,       1),       Cells(1,
gSourceColumn)).Find(what:="Fuel Quantity (gallons)").Column
    Case 2
        iFuelQuantityColumn       =       shSourceSheet.Range(Cells(1,       1),       Cells(1,
gSourceColumn)).Find(what:="Fuel Efficiency (mpg)").Column
    Case 3
        iFuelQuantityColumn       =       shSourceSheet.Range(Cells(1,       1),       Cells(1,
gSourceColumn)).Find(what:="Distance (miles)").Column
    End Select

    iFuelTypeColumn = shSourceSheet.Range(Cells(1, 1), Cells(1, gSourceColumn)).Find(what:="Fuel
Type").Column
    iWeightColumn = shSourceSheet.Range(Cells(1, 1), Cells(1, gSourceColumn)).Find(what:="Weight
(pounds)").Column

    If Dir(gOutPutFile, vbNormal) <> "" Then
        fnCalculateCO2 = False
        Exit Function
    End If
    Set wkResult = Workbooks.Add

    gCO2 = 0
    For i = 1 To gSourceRow - 1
        strFuelType = arSourceCellData(i, iFuelTypeColumn)
        If strFuelType <> "diesel" And strFuelType <> "gasoline" Then
            MsgBox "Warning!    Fuel Type is not diesel or gasoline! ", vbQuestion, "WARNING"
        End If
        garCO2(i)       =       fnCalculateData(Val(arSourceCellData(i,       iFuelQuantityColumn)),
Val(arSourceCellData(i, iWeightColumn)), strFuelType, gOptionType)
        gCO2 = gCO2 + garCO2(i)
    Next
```

```vba
    Set shResult = wkResult.Sheets.Add
    fnMakeSheet shSourceSheet.Cells(1, 1), shResult, garCO2()

    Dim k As Integer
    For j = 1 To giShowSheet

        iCount = 0
        iShowColumn = garShowColumn(j, 0)
        ReDim strTemp(garShowColumn(j, 0) - 1)
        ReDim arResultData(15000, iShowColumn + 2)

        strTitle = gvaTitle(1, garShowColumn(j, 1))

        arResultData(0, 0) = gvaTitle(1, garShowColumn(j, 1))
        For k = 2 To iShowColumn
            If gvaTitle(1, garShowColumn(j, k)) = "Origin" Then
                strTitle = strTitle & "," & "Route"
            ElseIf gvaTitle(1, garShowColumn(j, k)) = "Destination" Then
            Else
                strTitle = strTitle & "," & gvaTitle(1, garShowColumn(j, k))
            End If

            arResultData(0, k - 1) = gvaTitle(1, garShowColumn(j, k))
        Next
        If iShowColumn = 4 Then
            strTitle = "All"
        End If

        arResultData(0, iShowColumn) = "Total Emissions"
        arResultData(0, iShowColumn + 1) = "Total Shipment"
        arResultData(0, iShowColumn + 2) = "Average Emissions"

        shSourceSheet.Activate
        shSourceSheet.Sort.SortFields.Clear

        shSourceSheet.Sort.SortFields.Add        Key:=Range(Cells(2,        garShowColumn(j,        1)),
Cells(gSourceRow,        garShowColumn(j,        1))),        SortOn:=xlSortOnValues,        Order:=xlAscending,
DataOption:=xlSortNormal
        For k = 2 To iShowColumn
            shSourceSheet.Sort.SortFields.Add        Key:=Range(Cells(2,        garShowColumn(j,        k)),
Cells(gSourceRow,        garShowColumn(j,        k))),        SortOn:=xlSortOnValues,        Order:=xlAscending,
DataOption:=xlSortNormal
        Next
```

```vba
        shSourceSheet.Sort.SetRange        shSourceSheet.Range(Cells(2,    1),    Cells(gSourceRow,
gSourceColumn))
        shSourceSheet.Sort.Header = xlNo
        shSourceSheet.Sort.MatchCase = False
        shSourceSheet.Sort.Orientation = xlTopToBottom
        shSourceSheet.Sort.SortMethod = xlPinYin
        shSourceSheet.Sort.Apply

        arSourceCellData = shSourceSheet.Range(Cells(2, 1), Cells(gSourceRow, gSourceColumn))

        For i = 1 To gSourceRow - 1

            blCompare = True
            For k = 0 To iShowColumn - 1
                If strTemp(k) <> arSourceCellData(i, garShowColumn(j, k + 1)) Then
                    blCompare = False
                    Exit For
                End If
            Next
            If blCompare = False Then
                iCount = iCount + 1
                For k = 0 To iShowColumn - 1
                    strTemp(k) = arSourceCellData(i, garShowColumn(j, k + 1))
                    arResultData(iCount, k) = arSourceCellData(i, garShowColumn(j, k + 1))
                Next
            End If

            strFuelType = arSourceCellData(i, iFuelTypeColumn)
            arResultData(iCount,  iShowColumn)  =  Val(arResultData(iCount,  iShowColumn))  +
fnCalculateData(Val(arSourceCellData(i,          iFuelQuantityColumn)),          Val(arSourceCellData(i,
iWeightColumn)), strFuelType, gOptionType)

            arResultData(iCount, iShowColumn + 1) = Val(arResultData(iCount, iShowColumn + 1))
+ 1
            arResultData(iCount,  iShowColumn + 2) = Val(arResultData(iCount,  iShowColumn)) /
Val(arResultData(iCount, iShowColumn + 1))
        Next
        If strTitle = "Origin" Then strTitle = "Route"
        Set shResult = wkResult.Sheets.Add(After:=Worksheets(1))
        fnMakeSheet strTitle, shResult, arResultData
        Next

    ttp = Application.GetSaveAsFilename(gOutPutFile, fileFilter:="Excel Files (*.xlsx), *.xlsx")
    If ttp <> False Then gOutPutFile = ttp
```

```vba
        wkResult.SaveAs gOutPutFile
        wkResult.Close
        Set wkResult = Nothing
End Function


Function fnCalculateData(dbFuelorDistance As Double, dbWeight As Double, strFuelType As String, iOptionType As Integer) As Double


    Select Case iOptionType
    Case 1
        If strFuelType = "diesel" Then
            gfnCalculateData = dbFuelorDistance * 2.77 * 1 * 44 / 12 * 0.00045359237
        ElseIf strFuelType = "gasoline" Then
            fnCalculateData = dbFuelorDistance * 2.4 * 1 * 44 / 12 * 0.00045359237
        End If
    Case 2
        If strFuelType = "diesel" Then
            gfnCalculateData = dbWeight / dbFuelorDistance * 2.77 * 1 * 44 / 12 * 0.00045359237
        ElseIf strFuelType = "gasoline" Then
            fnCalculateData = dbWeight / dbFuelorDistance * 2.4 * 1 * 44 / 12 * 0.00045359237
        End If
    Case 3
         If strFuelType = "diesel" Then
            gfnCalculateData = dbFuelorDistance * dbWeight * 3200 / 139200 * 2.77 * 1 * 44 / 12 * 0.00045359237
        ElseIf strFuelType = "gasoline" Then
            fnCalculateData = dbFuelorDistance * dbWeight * 3200 / 125000 * 2.4 * 1 * 44 / 12 * 0.00045359237
        End If
    End Select
End Function
Function fnMakeSheet(strColumnTitle As String, shResult As Worksheet, vaCellData As Variant) As Boolean
    Dim arTitleData(1 To 1, 1 To 4) As String
    If strColumnTitle = "Shipment ID" Then
        arTitleData(1, 1) = "Total Emissions"
        arTitleData(1, 2) = "Total Shipment"
        arTitleData(1, 3) = "Average Emissions"
        arTitleData(1, 4) = ""
    Else
        arTitleData(1, 1) = "Analysis by " & strColumnTitle
        arTitleData(1, 2) = ""
        arTitleData(1, 3) = ""
        arTitleData(1, 4) = ""
```

```vba
        End If
        shResult.Activate

        If strColumnTitle = "Shipment ID" Then
            shResult.Name = "Analysis summary"
            shResult.Cells(1, 1).Resize(1, 4) = arTitleData
            shResult.Cells(2, 1) = gCO2
            shResult.Cells(2, 2) = UBound(garCO2)
            shResult.Cells(2, 3) = gCO2 / UBound(garCO2)
        Else
            shResult.Name = strColumnTitle
            shResult.Cells(1, 1).Resize(1, 4) = arTitleData
            shResult.Cells(2, 1).Resize(15000, UBound(vaCellData, 2) + 1) = vaCellData
        End If
        shResult.Columns.AutoFit
End Function


Function fnMakeResultFile(shSourceSheet As Worksheet) As Boolean

        Dim i                      As Integer
        Dim iLoop                  As Integer
        Dim strColumnTitle         As String
        Dim iDataColumn            As Integer
        Dim vaCellData             As Variant
        iLoop = 0

        vaCellData = shSourceSheet.Range(Cells(2, 1), Cells(gSourceRow, gSourceColumn))
        Do While iLoop < 6
            shSourceSheet.Activate

            iDataColumn = iLoop + 1
            strColumnTitle = garSourceColumnTitle(iLoop)
            Dim sxdata
            sxdata  =  shSourceSheet.Range(Cells(2,  11),  Cells(shSourceSheet.Cells(Rows.Count,
11).End(3).Row, 11))
            shSourceSheet.Range(Cells(1,  11),  Cells(shSourceSheet.Cells(Rows.Count,  11).End(3).Row,
11)) = ""
            shSourceSheet.Activate
            shSourceSheet.AutoFilterMode = False
            Dim fqc As Range
            Dim ftc As Range
            Dim iFuelQuantityColumn        As Integer
            Dim iFuelTypeColumn             As Integer
```

```vba
Dim CO2
Dim caldata(1 To 10000, 1 To 7)
Dim n            As Long
Dim cs
cs = 0
n = 1
For i = LBound(sxdata) To UBound(sxdata)
      For j = LBound(vaCellData, 1) To UBound(vaCellData, 1)
            If sxdata(i, 1) = vaCellData(j, iDataColumn) Then
                  Set  ftc = shSourceSheet.Range(Cells(1,  1),  Cells(1,  10)).Find(what:="Fuel
Type")
                        If Not ftc Is Nothing Then
                          iFuelTypeColumn = ftc.Column
                          End If
                          Set ftc = Nothing
                          Set    fqc    =      shSourceSheet.Range(Cells(1,      1),      Cells(1,
10)).Find(what:="Fuel Quantity (gallons)")
                              If Not fqc Is Nothing Then
                               iFuelQuantityColumn = fqc.Column
                               End If
                        Set fqc = Nothing

                        If vaCellData(j, iFuelTypeColumn) = "diesel" Then
                            CO2 = Val(vaCellData(j, iFuelQuantityColumn)) * 2.77 * 1 * 44 / 12
                            caldata(n, 1) = sxdata(i, 1)
                            caldata(n, 2) = caldata(n, 2) + CO2
                         End If
                        cs = cs + 1
                       ElseIf vaCellData(j, iFuelTypeColumn) = "gasoline" Then
                        CO2 = Val(vaCellData(j, iFuelQuantityColumn)) * 2.4 * 1 * 44 / 12
                           caldata(n, 1) = sxdata(i, 1)
                        caldata(n, 2) = caldata(n, 2) + CO2
                        cs = cs + 1
                        End If
                Next

         If cs <> 0 Then
               caldata(n, 3) = cs
               caldata(n, 4) = caldata(n, 2) / cs
               cs = 0
               n = n + 1
         End If
    Next
```

```vba
    Dim outdata(1 To 2, 1 To 4)
    If strColumnTitle = "Shipment ID" Then
        outdata(1, 1) = ""
        outdata(1, 2) = "Total Emissions"
        outdata(1, 3) = "Total Shipment"
        outdata(1, 4) = "Average Emissions"
    Else
        outdata(1, 1) = "Analysis by " & strColumnTitle
        outdata(1, 2) = ""
        outdata(1, 3) = ""
        outdata(1, 4) = ""
        outdata(2, 1) = strColumnTitle
        outdata(2, 2) = "Total Emissions"
        outdata(2, 3) = "Total Shipment"
        outdata(2, 4) = "Average Emissions"
    End If

    If Dir(gOutPutFile, vbNormal) = "" Then

    Dim wkResult As Workbook
    Set wkResult = Workbooks.Add
    wkResult.Activate
    With wkResult.Sheets.Add
      If strColumnTitle = "Shipment ID" Then
      Dim sm, st
.Name = "Analysis summery"
.Cells(1, 1).Resize(1, 4) = outdata
        .Cells(2, 1).Resize(100, 4) = caldata
      .Columns("A:A").EntireColumn.Delete
     sm = Application.Sum(Range("A2:A100"))
          st = Application.Sum(Range("B2:B100"))
     .Range("A2:C100") = ""
     .Cells(2, 1) = sm
     .Cells(2, 2) = st
        .Cells(2, 3) = sm / st
      Else
     .Name = "Analysis by " & strColumnTitle
        .Cells(1, 1).Resize(2, 4) = outdata
        .Cells(3, 1).Resize(100, 4) = caldata
End If
.Columns.AutoFit
End With
wkResult.SaveAs gOutPutFile
wkResult.Close
```

```
        Else
            With Workbooks.Open(gOutPutFile)
              .Activate
              sc = .Sheets.Count - 2
              With .Sheets.Add
              .Move After:=Sheets(sc)

                .Name = "Analysis by " & strColumnTitle
                    .Cells(1, 1).Resize(2, 4) = outdata
                    .Cells(3, 1).Resize(100, 4) = caldata
              .Columns.AutoFit
              End With
              .Save
              .Close
              End With
            End If
            iLoop = iLoop + 1
            Erase caldata
            Erase outdata
        Loop

        fnMakeResultFile = True
End Function
```