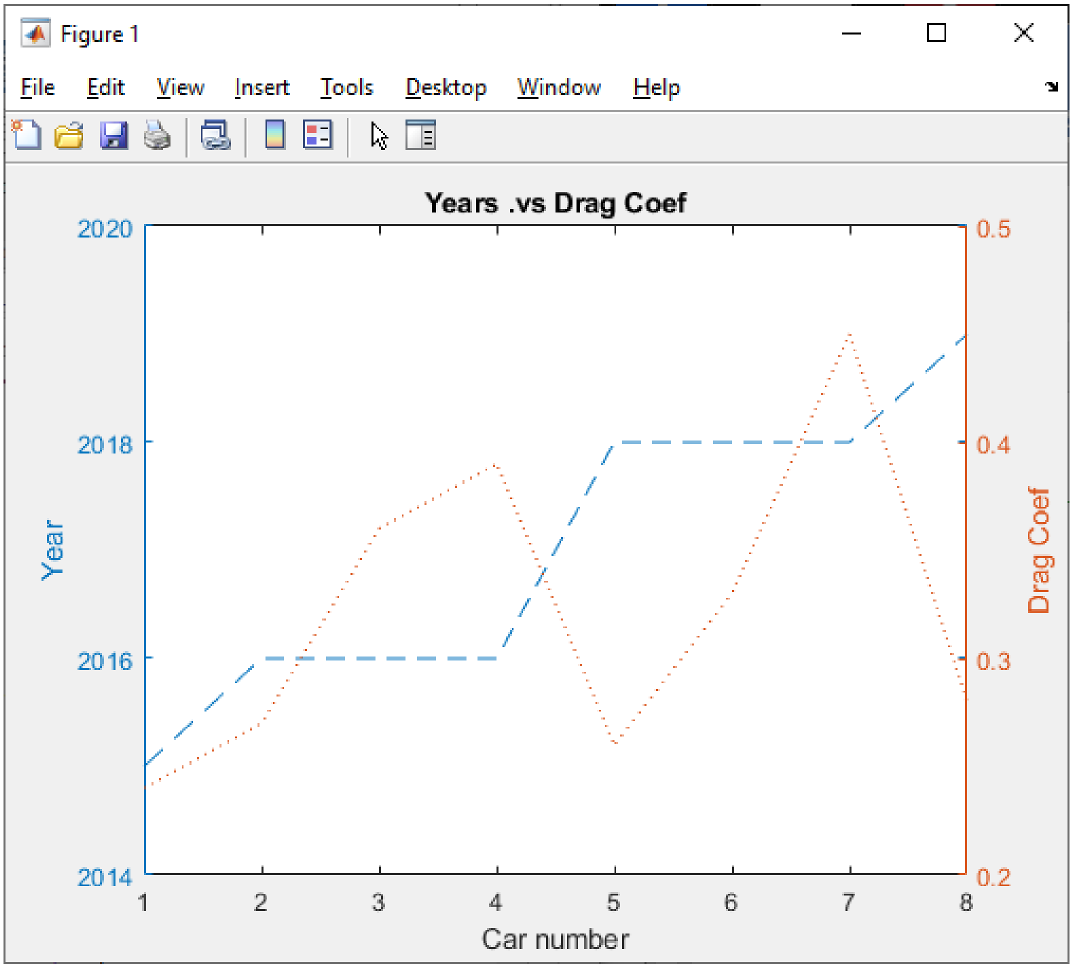
Grading input data file

dragCoef07.txt

```
2015   0.24
2016   0.27
2016   0.36
2016   0.39
2018   0.26
2018   0.33
2018   0.45
2019   0.28
```

Grading output data file

```
        All Cars
Year      Drag    Frontal
          Coef    Area(m^2)
2015      0.24    3.8095 *
2016      0.27    3.3862 *
2016      0.36    2.5397
2016      0.39    2.3443
2018      0.26    3.5165 *
2018      0.33    2.7706
2018      0.45    2.0317
2019      0.28    3.2653 *
* Frontal area above area
```



```
% J Hundley
% assign07.m
% compute and display the frontal area of the car drag coefficients

clc, clear all

% ***** CONSTANT *****
FILENAME = 'dragCoef07.txt';
DENSITY  = 1.225;    % kg/m3,density of air at sea level
FORCE    = 350;      % Newtons (N), force of the drag
VELOCITY = 90/3.6;   % or 25 m/s, velocity of object relative to the fluid

% autonomous function to compute frontal area
getArea  = @(coef) (2*FORCE) ./ ( coef * DENSITY * VELOCITY ^ 2 );
% ***** INPUT *****
% open data file
[ fileID, msg ] = fopen( FILENAME, 'r' );
% does file exist?
if fileID <= 0
    fprintf( 'File not available\n' )
else
    % file exist continue
    % read year and drag coefficient from file
    car = 0;
    while ~feof( fileID )
        car = car + 1;
        yearDrag( car, 1 ) = fscanf( fileID, '%f', 1 ); % year
        yearDrag( car, 2 ) = fscanf( fileID, '%f', 1 ); % drag coefficient
    end

    % ***** COMPUTE *****
    % compute the frontal area of all cars
    allArea = getArea( yearDrag(:,2) );

    % add frontal area as third column of matrix
    yearDrag(:,3) = allArea';
    % sort matrix by year column ascending order
    sortMatrix    = sortrows( yearDrag, 1 );
    % get average frontal area
    aveArea       = mean( sortMatrix(:,3) );

    % ***** OUTPUT *****
    % print table again using fprintf()
    printReport( sortMatrix, aveArea )

    fclose( fileID );
end
```

```
% =================================================== PRIMARY FUNCTION FILE printReport.m
% J Hundley
% printReport.m used by assign07.m
function  [] = printReport( carMatrix, aveArea )
% % print report and draw graph

% print rtable
fprintf( '        All Cars\n' )
fprintf( 'Year      Drag    Frontal\n' )
fprintf( '          Coef    Area(m^2)\n' )
for car = 1:length(carMatrix(:,1) )
    fprintf( '%4.0f      %4.2f    %6.4f', carMatrix(car,:) )
    if carMatrix(car,3) > aveArea
        fprintf( ' *\n' )
    else
        fprintf( '\n' )
    end
end
fprintf( '* Frontal area above area \n' )

% draw a plot of the years and drag coef
plotData( carMatrix( :,1 ), carMatrix( :,2 ) )

end


% --------------------------- SUBFUNCTION FUNCTION LOCATED IN printReport.m WHICH USES IT
function [] = plotData( years, dragCoef )
% draw a plot of the years and drag coef
numCars = 1:length( years );
[hAx,hLine1,hLine2] = plotyy( numCars, years, numCars, dragCoef );
title( 'Years .vs Drag Coef' )
xlabel( 'Car number' )
hLine1.LineStyle = '--';
hLine2.LineStyle = ':';
ylabel(hAx(1),'Year')      % left y-axis
ylabel(hAx(2),'Drag Coef') % right y-axis
end
```

*Read **all** instructions*
before beginning your work.

COMP1200-MATLAB – assign07
Due 4:45pm – MONDAY 23, 2020
Submit `assign07.m` and
`printReport.m` **via Canvas**

NOTE:
Your submitted file(s) MUST be
lled and cased as instructed.

NEW DATE

Before you start writing your program:

Read the complete instructions. Write an algorithm to use to as comments in you script.
An **algorithm** contains the steps needed to guide you through solving a problem.

***Program*: assign07.m**

Your assign07.m will read the data file and write the same report as assign06 with the follow modifications.

Use **`feof()`** to control the read file loop.
Use an anonymous function to compute the frontal area.
Use a user-defined function to print the report and draw a graph.
Use a subfunction to draw the graph.

In assign07.m

Open file and read data into a two column matrix.
Use the anonymous function given below to compute the frontal area. Add area as third matrix column and sort the matrix by year.
Get the average frontal area.
Use a user-defined function to print the report and draw a graph.

An anonymous function should be place near the top of the script with the constants.

```
% autonomous function to compute frontal area
getArea = @(coef) (2*FORCE) ./ ( coef * DENSITY * VELOCITY ^ 2 );
```

Do not use commands and statements beyond what has been taught on class.

New commands:
Use `feof()` to end the read file loop.
`plotyy()`
Only continue if file is available otherwise
Continue:
Use functions from previous assignments as needed.
`fscanf()` to read the data file
Only continue if file is available.
Use `fprintf()` for all output.
Use descriptive variables.

In **`printReport()`** include all statements and comments needed to print all the output and draw a graph.

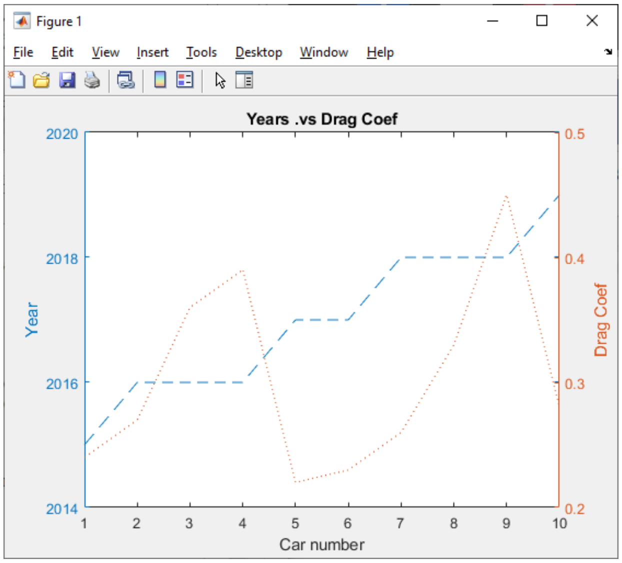
Use the given subfunction to plot the years and drag coefs.

The function should be named as given and save in a file **`printReport.m`**.
Variable names may be different, but the order and quantity should be as given.

```
function [] = printReport( carMatrix, aveArea )
% print report and draw graph
```

Type **`plotData()`** as subfunction located in the primary function, **`printReport.m`** file. The subfunction is used by the primary function, **`printReport()`** , to draw a graph. Colors may differ.

```
function [] = plotData( years, dragCoef )
% draw a plot of the years and drag coef
numCars = 1:length( years );
[hAx,hLine1,hLine2] = plotyy( numCars,
years, numCars, dragCoef );
title( 'Years .vs Drag Coef' )
xlabel( 'Car number' )
hLine1.LineStyle = '--';
hLine2.LineStyle = ':';
ylabel(hAx(1),'Year')      % left y-axis
ylabel(hAx(2),'Drag Coef') % right y-axis
end
```



***Problem* CONSTANTS:** `(with units)`

filename = `'dragCoef07.txt'`

velocity, horizontal_force, air_density

***Problem Inputs*:** (with units)

drag coefficient

Problem Outputs: (with units)

frontal area

Other variables: (with units)

average frontal area

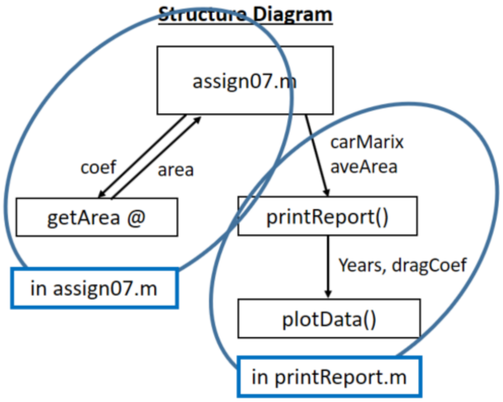
Equation:

See above.

Algorithm:

Using the following section comments and previous assignment files as a guide create an algorithm for the current requirements. Use the algorithm as comments in your `assign07.m`

```
% ***** CONSTANT *****
% ***** INPUT *****
% ***** COMPUTE *****
% ***** OUTPUT *****
```



Instructions for all assignment scripts:

- □ See Standards for Documentation of MATLAB Programs on the Canvas Resources page.
- □ Insert comments at the top and throughout each file.
  - o Include the follow comments at the beginning of this (and ALL) files.
    - % submitter's name, **GROUP # or "none"**
    - % other group members' names or **"none"**
    - % `program file name`, ex. assign02a.m
    - % due date of the assignment
    - % **statement about collaboration REQUIRED.**
    - % a short narrative about what the file does
  - o Use the algorithm given as comments throughout your program.
- □ Observe the instructor's rule for naming variables.
  - o Use ALL CAPS for constants variable names.
  - o Start other variables with lower case.
  - o Use descriptive variable names.
- □ Use Sample Input/Output as a guide.
- □ Code clarity:
  - o Indent blocks as needed. **Use Smart Indent.**
  - o Divide your solution program code into sections as noted in the algorithm. Use blank lines as needed to group statements.
  - o Use section comments as well as the algorithm step comments.
  - o Remove statements from previous assignments that do not apply to the current requirements.
- □ Use comments to show units.
- □ **Use the CONSTANT and variable names, not numbers.**
  Exceptions are incrementers (or counters) and numbers without identity.
- □ No extra output, i.e. use semicolons

GRADE OF ZERO **for a file if submitter name not part of Canvas group.**

(-3pts) No CURRENT GROUP# or "none".
(-3pts) For your own protection, type "none" for other group members if submitting alone.
(-5pts) Five point penalty for not joining your Canvas group.

(-5pts) Starting with assign06, penalty applied for omitting the name of any group member from a script comment list or an incomplete name of a group member in a script comment list. This penalty will be applied to the group grade if at least one file has incomplete or incorrect name information.

(-5pts) Zero points for comments if no collaboration statement.

Submit via Canvas:

`assign07.m`	MATLAB script file
`printReport.m`	user-defined function file

NOTE: Your submitted file(s) MUST be spelled and cased as instructed.
One submission per group. Canvas links members to files and rubric.
A script cannot run from Canvas. It must be downloaded, saved, and "run".