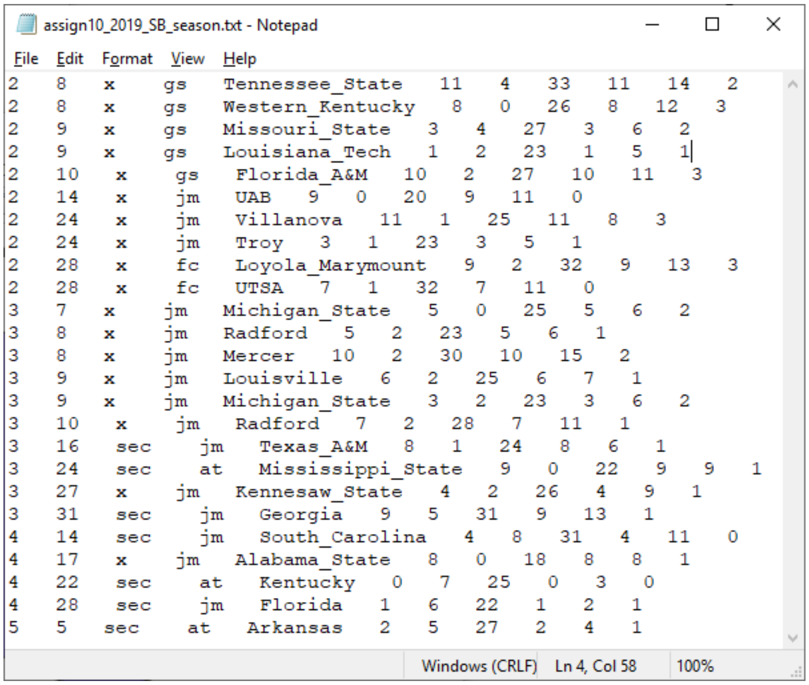
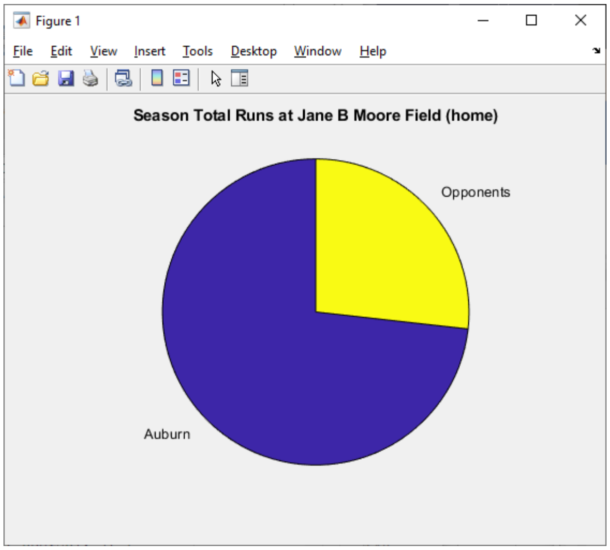
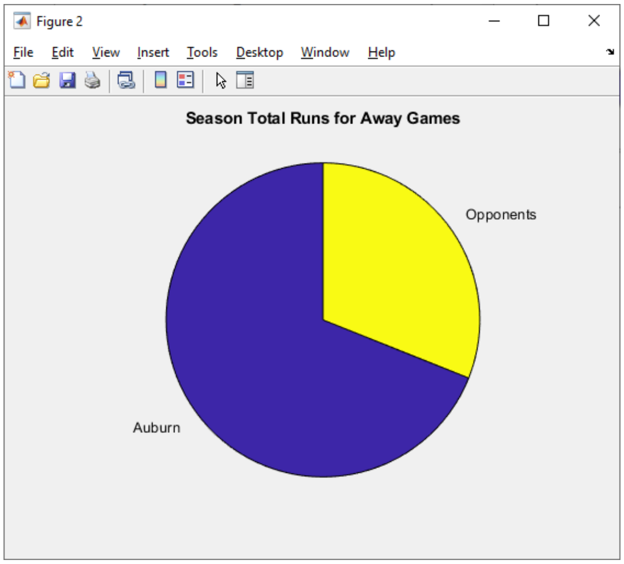

assign10_2019_SB_season.txt - Notepad
File Edit Format View Help
Windows (CRLF)
Ln 4, Col 58
100%

```
2   8   x   gs   Tennessee_State   11   4   33   11   14   2
2   8   x   gs   Western_Kentucky   8   0   26   8   12   3
2   9   x   gs   Missouri_State   3   4   27   3   6   2
2   9   x   gs   Louisiana_Tech   1   2   23   1   5   1
2   10   x   gs   Florida_A&M   10   2   27   10   11   3
2   14   x   jm   UAB   9   0   20   9   11   0
2   24   x   jm   Villanova   11   1   25   11   8   3
2   24   x   jm   Troy   3   1   23   3   5   1
2   28   x   fc   Loyola_Marymount   9   2   32   9   13   3
2   28   x   fc   UTSA   7   1   32   7   11   0
3   7   x   jm   Michigan_State   5   0   25   5   6   2
3   8   x   jm   Radford   5   2   23   5   6   1
3   8   x   jm   Mercer   10   2   30   10   15   2
3   9   x   jm   Louisville   6   2   25   6   7   1
3   9   x   jm   Michigan_State   3   2   23   3   6   2
3   10   x   jm   Radford   7   2   28   7   11   1
3   16   sec   jm   Texas_A&M   8   1   24   8   6   1
3   24   sec   at   Mississippi_State   9   0   22   9   9   1
3   27   x   jm   Kennesaw_State   4   2   26   4   9   1
3   31   sec   jm   Georgia   9   5   31   9   13   1
4   14   sec   jm   South_Carolina   4   8   31   4   11   0
4   17   x   jm   Alabama_State   8   0   18   8   8   1
4   22   sec   at   Kentucky   0   7   25   0   3   0
4   28   sec   jm   Florida   1   6   22   1   2   1
5   5   sec   at   Arkansas   2   5   27   2   4   1
```





```
Season Average Scores by Category
                  Home        Away
                AU    Opp   AU    Opp
SEC games      5.50  5.00  3.67  4.00
nonSEC games   6.45  1.27  7.00  2.14

Season Highest Scores and Game Dates by Category
                   Home                Away
              Auburn  Opponent   Auburn   Opponent
SEC games      9 03/31  8 04/14   9 03/24  7 04/22
nonSEC games 11 02/24  2 03/08  11 02/08  4 02/08
```

```
% J Hundley
% assign10.m
% April 17, 2020
% read AU softball stats and print season report
clc, clear all
%***** CONSTANT ***********************************************************
SB_STATS = 'assign10_2019_SB_season.txt';

%***** INPUT ***************************************************************
% is file available?
if ~exist( SB_STATS, 'file' )
    disp( 'File not available' )
else
    % files available
    % read SB stats amd abbreviations files
    [ dates(:,1),dates(:,2), conf,loc,opp, scores(:,1),scores(:,2) ] =...
        textread( SB_STATS, '%f%f %s%s%s %f%f %*f%*f%*f%*f' );
```

```
% SEASON AVERAGE SCOREES BY CATEGORY VARIABLES
secJmNumGames  = 0;  % sec home
secJmSumAu     = 0;
secJmSumOp     = 0;
secAwNumGames  = 0;  % sec away
secAwSumAu     = 0;
secAwSumOp     = 0;
nSecJmNumGames = 0;  % non-sec home
nSecJmSumAu    = 0;
nSecJmSumOp    = 0;
nSecAwNumGames = 0;  % non-sec away
nSecAwSumAu    = 0;
nSecAwSumOp    = 0;

% GET INFO FOR SEASON AVERAGE SCORES BY CATEGORY
%   num games, AU total runs, Opp total runs
% for next section, build vectors for each category
numGames = length( dates(:,1) );
for g = 1 : numGames
    if ~isempty( strfind( conf{ g }, 'sec' ) )
        % sec
        if ~isempty( strfind( loc{ g }, 'jm' ) )
            % sec home
            secJmNumGames = secJmNumGames + 1;
            secJmSumAu    = secJmSumAu    + scores( g,1 );
            secJmSumOp    = secJmSumOp    + scores( g,2 );
            % build matrix au op sec home scores and game #
            secJmRunsAu(secJmNumGames,1) = scores( g,1 );
            secJmRunsAu(secJmNumGames,2) = g;
            secJmRunsOp(secJmNumGames,1) = scores( g,2 );
            secJmRunsOp(secJmNumGames,2) = g;
        else
            % sec away
            secAwNumGames = secAwNumGames + 1;
            secAwSumAu    = secAwSumAu    + scores( g,1 );
            secAwSumOp    = secAwSumOp    + scores( g,2 );
            % build matrix au op sec away scores and game #
            secAwRunsAu(secAwNumGames,1) = scores( g,1 );
            secAwRunsAu(secAwNumGames,2) = g;
            secAwRunsOp(secAwNumGames,1) = scores( g,2 );
            secAwRunsOp(secAwNumGames,2) = g;
        end
    else
        % nonSec
        if ~isempty( strfind( loc{ g }, 'jm' ) )
            % nonSec home
            nSecJmNumGames = nSecJmNumGames + 1;
            nSecJmSumAu   = nSecJmSumAu   + scores( g,1 );
            nSecJmSumOp   = nSecJmSumOp   + scores( g,2 );
            % build matrix au op non-sec home scores and game #
            nSecJmRunsAu(nSecJmNumGames,1) = scores( g,1 );
            nSecJmRunsAu(nSecJmNumGames,2) = g;
            nSecJmRunsOp(nSecJmNumGames,1) = scores( g,2 );
            nSecJmRunsOp(nSecJmNumGames,2) = g;
        else
            % nonSec away
            nSecAwNumGames = nSecAwNumGames + 1;
            nSecAwSumAu   = nSecAwSumAu   + scores( g,1 );
            nSecAwSumOp   = nSecAwSumOp   + scores( g,2 );
            % build matrix au op non-sec away scores and game #
            nSecAwRunsAu(nSecAwNumGames,1) = scores( g,1 );
            nSecAwRunsAu(nSecAwNumGames,2) = g;
            nSecAwRunsOp(nSecAwNumGames,1) = scores( g,2 );
            nSecAwRunsOp(nSecAwNumGames,2) = g;
        end
    end
    end

    % GET INFO FOR SEASONHIGHEST SCORES & GAMES DATES BY CATEBORY VARIABLES
    % sec home away: max points and game #
    [  secJmMaxAu, loc ] = max(  secJmRunsAu(:,1) );
    secJmMaxAuGame  = secJmRunsAu( loc,2 );
    [  secJmMaxOp, loc ] = max(  secJmRunsOp(:,1) );
    secJmMaxOpGame  = secJmRunsOp( loc,2 ) ;
    [  secAwMaxAu, loc ] = max(  secAwRunsAu(:,1) );
    secAwMaxAuGame  = secAwRunsAu( loc,2 );
    [  secAwMaxOp, loc ] = max(  secAwRunsOp(:,1) );
    secAwMaxOpGame  = secAwRunsOp( loc,2 );
    % non-sec home away: max points and game #
    [ nSecJmMaxAu, loc ] = max( nSecJmRunsAu(:,1) );
    nSecJmMaxAuGame = nSecJmRunsAu(loc,2);
    [ nSecJmMaxOp, loc ] = max( nSecJmRunsOp(:,1) );
    nSecJmMaxOpGame = nSecJmRunsOp(loc,2);
    [ nSecAwMaxAu, loc ] = max( nSecAwRunsAu(:,1) );
    nSecAwMaxAuGame = nSecAwRunsAu(loc,2);
    [ nSecAwMaxOp, loc ] = max( nSecAwRunsOp(:,1) );
    nSecAwMaxOpGame = nSecAwRunsOp(loc,2);

    % GET INFO FOR CHARTS
    % total runs by category
    totalAuJm = secJmSumAu + nSecJmSumAu;
    totalOpJm = secJmSumOp + nSecJmSumOp;
    totalAuAw = secAwSumAu + nSecAwSumAu;
    totalOpAw = secAwSumOp + nSecAwSumOp;

    % ***** OUTPUT *********************************************************
    % 1 -- season average scores
    fprintf( '\nSeason Average Scores by Category\n' )
    fprintf( '                  Home        Away\n' )
    fprintf( '                AU    Opp   AU    Opp\n' )
    fprintf( 'SEC games     %5.2f %5.2f %5.2f %5.2f \n',...
        secJmRunsAu/secJmNumGames, secJmRunsOp/secJmNumGames,...
        secAwRunsAu/secAwNumGames, secAwRunsOp/secAwNumGames )
    fprintf( 'nonSEC games  %5.2f %5.2f %5.2f %5.2f \n',...
        nSecJmRunsAu/nSecJmNumGames, nSecJmRunsOp/nSecJmNumGames,...
        nSecAwRunsAu/nSecAwNumGames, nSecAwRunsOp/nSecAwNumGames )

    % 2 -- season highest points
    fprintf( '\nSeason Highest Scores and Game Dates by Category\n' )
    fprintf( '                   Home               Away\n' )
    fprintf( '              Auburn  Opponent   Auburn   Opponent\n' )
    fprintf( 'SEC games    %2d %02d/%02d %2d %02d/%02d  %2d %02d/%02d %2d %02d/%02d \n',...
        secJmMaxAu, dates(secJmMaxAuGame,1), dates(secJmMaxAuGame,2),...
        secJmMaxOp, dates(secJmMaxOpGame,1), dates(secJmMaxOpGame,2),...
        secAwMaxAu, dates(secAwMaxAuGame,1), dates(secAwMaxAuGame,2),...
        secAwMaxOp, dates(secAwMaxOpGame,1), dates(secAwMaxOpGame,2) )
    fprintf( 'nonSEC games %2d %02d/%02d %2d %02d/%02d  %2d %02d/%02d %2d %02d/%02d \n',...
        nSecJmMaxAu, dates(nSecJmMaxAuGame,1), dates(nSecJmMaxAuGame,2),...
        nSecJmMaxOp, dates(nSecJmMaxOpGame,1), dates(nSecJmMaxOpGame,2),...
        nSecAwMaxAu, dates(nSecAwMaxAuGame,1), dates(nSecAwMaxAuGame,2),...
        nSecAwMaxOp, dates(nSecAwMaxOpGame,1), dates(nSecAwMaxOpGame,2) )

    % PIE GRAPHS FOR TOTAL RUNS HOME AND TOTAL RUNS AWAY AUBURN AND OPPONENTS TOTAL RUNS
    figure(1)
    pie( [ totalAuJm, totalOpJm ], { 'Auburn','Opponents' } )
    title( 'Season Total Runs at Jane B Moore Field (home)' )

    figure(2)
    pie( [ totalAuAw, totalOpAw ], { 'Auburn','Opponents' } )
    title( 'Season Total Runs for Away Games' )
end
```

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

COMP1200-MATLAB – assign10
Due 4:45pm – Friday – April 17, 2020
Submit `assign10.m` and **`?????.m**`**
via Canvas
**** Using one or more user-defined functions is optional.**
If user-defined functions are used, student is responsible for submitting all files needed to produce the required output.

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

***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.

Problem:

Please note that due to the 2020 Auburn softball season ending early, the data for this assignment is for the 2019 Auburn softball season. Keep in mind that you do not know the number of games in the file. Let MATLAB give you this information.

***Program*: assign10.m**

Your assign10.m will read the data file, write a report, and draw a graph. Your script may be long with some repetition OR you may find a way to combine tasks getting some information. After the data is read into matrices, you can look through the games with the needed information.

See the season reports below for the information that is needed. For this assignment, you will determine several season statistics for Auburn and opponents at home and away games. I suggest starting by listing all the stats that you need in each category. Your script will look through the games for games that fit each category.

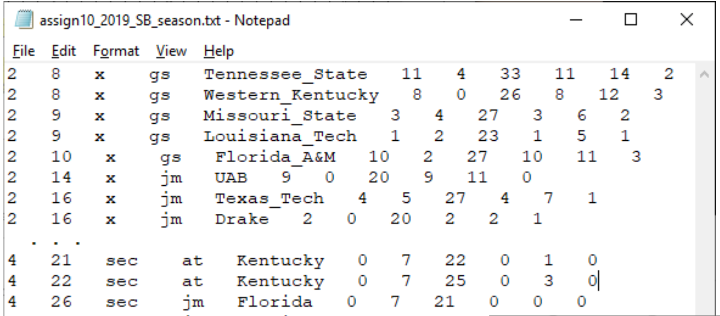
There are two pie charts in two separate figures showing the Auburn and opponents total runs at home and away. Each wedge should be labeled with 'Auburn' or 'Opponent'. See the charts.

New commands:
Only continue if file is available otherwise ONLY print message.
`pie(), figure()`
Functions to consider
`isempty(), strfind(), contains()`
If you do not have MATLAB 'help," search for a function at https://www.mathworks.com for assistance.

Continue:

***Sample Input**:*
Some games have been removed to shorten display.
Note: grading data file will not contain the same amount of game

```
                                               scores
month day sec home(jm) opponent                AU  opp   [      stats      ]
```

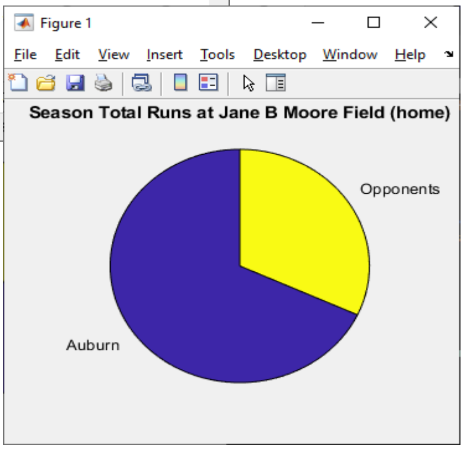


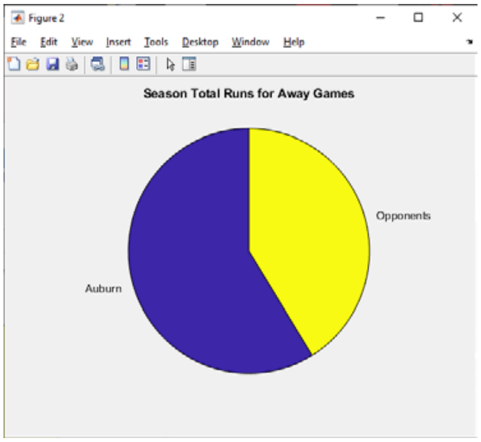
```
2   8   x    gs   Tennessee_State   11   4   33   11   14   2
2   8   x    gs   Western_Kentucky   8   0   26   8   12   3
2   9   x    gs   Missouri_State   3   4   27   3   6   2
2   9   x    gs   Louisiana_Tech   1   2   23   1   5   1
2   10   x    gs   Florida_A&M   10   2   27   10   11   3
2   14   x    jm   UAB   9   0   20   9   11   0
2   16   x    jm   Texas_Tech   4   5   27   4   7   1
2   16   x    jm   Drake   2   0   20   2   2   1
  . . .
4   21   sec    at   Kentucky   0   7   22   0   1   0
4   22   sec    at   Kentucky   0   7   25   0   3   0
4   26   sec    jm   Florida   0   7   21   0   0   0
4   27   sec    jm   Florida   2   1   34
4   28   sec    jm   Florida   1   6   22
5   3   sec   at   Arkansas   0   1   27
5   4   sec   at   Arkansas   1   7   24
5   5   sec   at   Arkansas   2   5   27
```

***Sample Output**:*
Zero fill dates with / between month and day.

```
Season Average Scores by Category
                 Home        Away
               AU     Opp   AU     Opp
SEC games      4.75  5.17  3.50  4.83
nonSEC games   6.89  1.37  7.50  2.40

Season Highest Scores and Game Dates by Category
                  Home                  Away
             Auburn  Opponent    Auburn    Opponent
SEC games    12 03/17 10 04/13  10 04/06  8 04/07
nonSEC games 15 02/17  5 02/16  12 03/13  8 03/01
```





To help you check you numbers:
Home 188 & 88
Away 117 & 82
Colors may vary on charts.

***Problem* CONSTANTS:** `(with units)`
filename = `'assign10_2019_SB_season.txt'`
***Problem Inputs*:** (with units)
  dates, conference, location, opponents, scores
Problem Outputs: (with units)
  See sample output
Other variables: (with units)
  As needed
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 `assign10.m`

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:

`assign10.m`	MATLAB script file
`???????.m**`	user-defined function file

**** Using one or more user-defined functions is optional.**
If user-defined functions are used, student is responsible for submitting all files needed to produce the required output.

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".