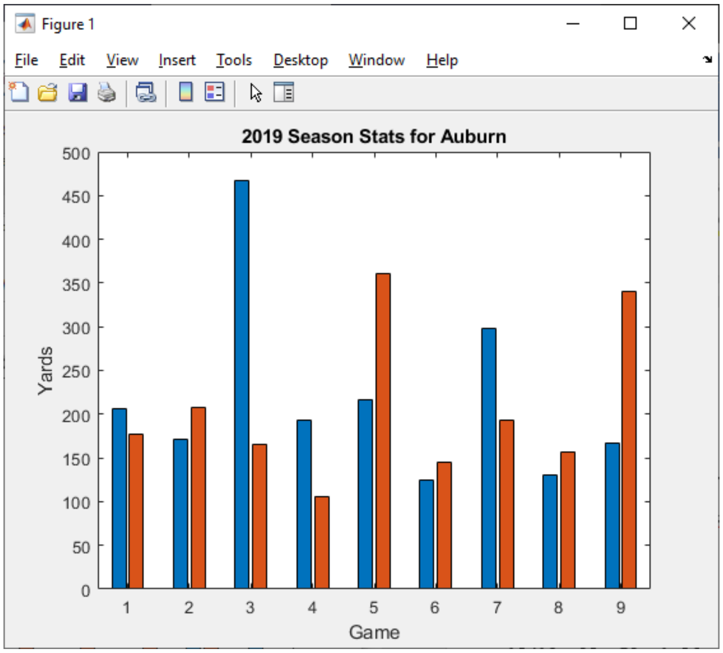
8

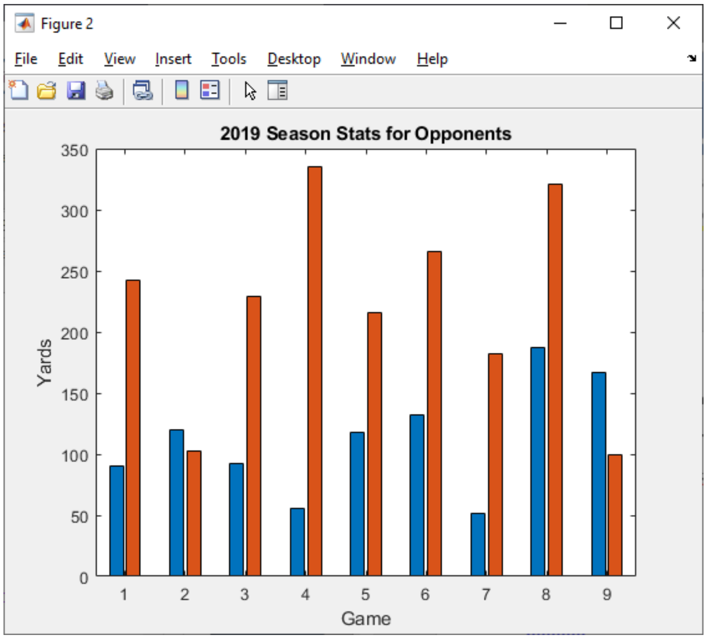
```
% J Hundley
% assign08.m
% using user-defined functions
% read season football stats from two files, Auburn and Opponent
% create a report and graph
clc, clear all8
format compact
%***** CONSTANTS *****
AU_STATS  = 'AU_stats08.txt';
OPP_STATS = 'opp_stats08.txt';

%***** INPUT *****
if ~exist( AU_STATS, 'file' ) && ~exist( OPP_STATS, 'file' )
    disp( 'one or both files are not available' )
else
    % files available continue with problem solution
    % read auburn and opponent: date, rushing, passing
    [ auDate(:,1), auDate(:,2),...                        % can be on long line
        auRush(:,1),auRush(:,2),auRush(:,3),auRush(:,4),...
        auPass(:,1),auPass(:,2),auPass(:,3),auPass(:,4) ]...
        = textread( AU_STATS, '%f%f %f%f%f%f %f%f%f%f' );  % spaces not required
    [ oppDate(:,1), oppDate(:,2),...
        oppRush(:,1),oppRush(:,2),oppRush(:,3),oppRush(:,4),...
        oppPass(:,1),oppPass(:,2),oppPass(:,3),oppPass(:,4) ]...
        = textread( OPP_STATS, '%f%f %f%f%f%f %f%f%f%f' );

    %***** COMPUTE *****
    % compute auburn & opponent rushing & passing yards per attemp for each game
    auRush(:,5)  = auRush(:,2)  ./ auRush(:,1);
    oppRush(:,5) = oppRush(:,2) ./ oppRush(:,1);
    auPass(:,5)  = auPass(:,2)  ./ auPass(:,1);
    oppPass(:,5) = oppPass(:,2) ./ oppPass(:,1);

    %***** OUTPUT *****
    % print stat report and draw graph
    outputReportGraph( auDate, auRush, auPass, 'Auburn', 1 )
    outputReportGraph( oppDate, oppRush, oppPass, 'Opponents', 2 )
end
```

```
% J Hundley                                                 SAVED IN outputReportGraph.m
% assign08.m
function [] = outputReportGraph( date, rushing, passing, name, figureNum )
% print stat report and draw graph
% print title, column headers
fprintf( '\n2019 Season Stats for %s \n', name )
fprintf( '          Rushing          Passing \n' )
fprintf( 'Date  Att Yds per Att  Att Yds per Att \n' )
for g = 1:length(date(:,1))
    fprintf( '%02.0f/%02.0f ', date(g,1), date(g,2) )
    fprintf( '%3.0f %3.0f %5.2f    ', rushing(g,1), rushing(g,2), rushing(g,5) )
    fprintf( '%3.0f %3.0f  %5.2f  \n', passing(g,1), passing(g,2), passing(g,5) )
end

% draw grouped bar chart of rushing and passing yards
chartTitle = sprintf( '2019 Season Stats for %s', name  );
figure( figureNum )
bar( [rushing(:,2), passing(:,2)] )
title( chartTitle )
xlabel( 'Game' )
ylabel( 'Yards' )
end
```

Output as of last game played file used by graders

```
2019 Season Stats for Auburn
          Rushing           Passing
Date  Att Yds per Att   Att Yds per Att
08/31  43 206  4.79      13 177  13.62
09/07  45 172  3.82      19 207  10.89
09/14  62 467  7.53      13 166  12.77
09/21  42 193  4.60      13 106   8.15
09/28  45 217  4.82      18 361  20.06
10/05  34 124  3.65      11 145  13.18
10/19  51 298  5.84      13 193  14.85
10/26  33 130  3.94      15 157  10.47
11/02  46 167  3.63      30 340  11.33

2019 Season Stats for Opponents
          Rushing           Passing
Date  Att Yds per Att   Att Yds per Att
08/31  33  90  2.73      28 242   8.64
09/07  30 120  4.00      10 103  10.30
09/14  32  92  2.88      18 229  12.72
09/21  21  56  2.67      31 335  10.81
09/28  38 118  3.11      13 216  16.62
10/05  33 132  4.00      25 266  10.64
10/19  28  52  1.86      19 182   9.58
10/26  46 187  4.07      32 321  10.03
11/02  42 167  3.98      14  99   7.07
```





Read all instructions before beginning your work.

COMP1200-MatLab - assign 08
Due 4:45 pm – Friday – November 8, 2019
Submit `assign08.m` and
`outputReportGraph.m` **via Canvas**

NOTE:
Your submitted file(s) MUST be spelled and cased as instructed.
[-5 points for not doing so.]

***Before you start writing your program**:*
Read the complete instructions.

***Program**:* **assign08.m**
Rushing and passing statistics are available for Auburn and the opponents for each football game. Read the statistics from files and print a report and graph of information as instructed.

***Problem Constants**:*
AU file name `'AU_stats08.txt'`
opp file name `'opp_stats08.txt'`

***Problem Inputs**:*
For Auburn and opponents:
game date in 2-col matrix
rushing stats in 4-col matrix
passing stats in 4-col matrix

Problem Outputs:
For Auburn's and opponents' reports (2):
- game date, print with leading zeros and `/`
- rushing attempts, yards, yards per attempt
- passing attempts, yards, yards per attempt

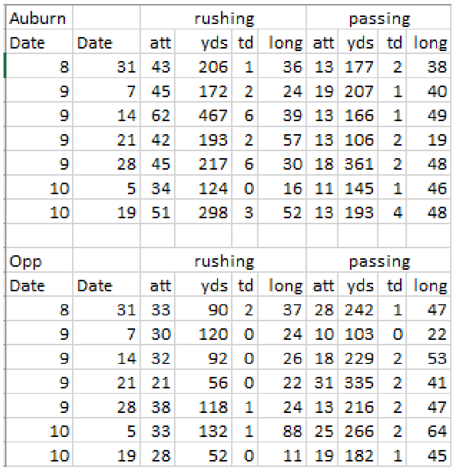
For Auburn's and opponents' graphs (2):
- rushing and passing yards

Other variables:
as needed

Equations:
as needed.

Output:

```
2019 Season Stats for Auburn
          Rushing          Passing
Date  Att Yds per Att  Att Yds per Att
08/31  43 206  4.79     13 177  13.62
09/07  45 172  3.82     19 207  10.89
09/14  62 467  7.53     13 166  12.77
09/21  42 193  4.60     13 106   8.15
09/28  45 217  4.82     18 361  20.06
10/05  34 124  3.65     11 145  13.18
10/19  51 298  5.84     13 193  14.85

2019 Season Stats for Opponents
          Rushing          Passing
Date  Att Yds per Att  Att Yds per Att
```



Auburn		rushing				passing			
Date	Date	att	yds	td	long	att	yds	td	long
8	31	43	206	1	36	13	177	2	38
9	7	45	172	2	24	19	207	1	40
9	14	62	467	6	39	13	166	1	49
9	21	42	193	2	57	13	106	2	19
9	28	45	217	6	30	18	361	2	48
10	5	34	124	0	16	11	145	1	46
10	19	51	298	3	52	13	193	4	48
Opp		rushing				passing			
Date	Date	att	yds	td	long	att	yds	td	long
8	31	33	90	2	37	28	242	1	47
9	7	30	120	0	24	10	103	0	22
9	14	32	92	0	26	18	229	2	53
9	21	21	56	0	22	31	335	2	41
9	28	38	118	1	24	13	216	2	47
10	5	33	132	1	88	25	266	2	64
10	19	28	52	0	11	19	182	1	45

Do not use commands and statements beyond what has been taught on class.
New commands:
Only continue if both files exist.
Reading from two files using two `textread()` into matrices as described in Input using colon notation for columns.
Use data to determine number of games.
In a counting loop, use three `fprintf()` statements for each line of numbers.
`sprintf()` to build title string for graphs
`figure()`
grouped `bar()` chart with `title()`, `xlabel()`,`ylabel()`

Continue to use:
Use given function definition, twice.
Set decimals as shown with columns right aligned.
Use descriptive variables.

```
08/31  33  90  2.73     28 242   8.64
09/07  30 120  4.00     10 103  10.30
09/14  32  92  2.88     18 229  12.72
09/21  21  56  2.67     31 335  10.81
09/28  38 118  3.11     13 216  16.62
10/05  33 132  4.00     25 266  10.64
10/19  28  52  1.86     19 182   9.58
```

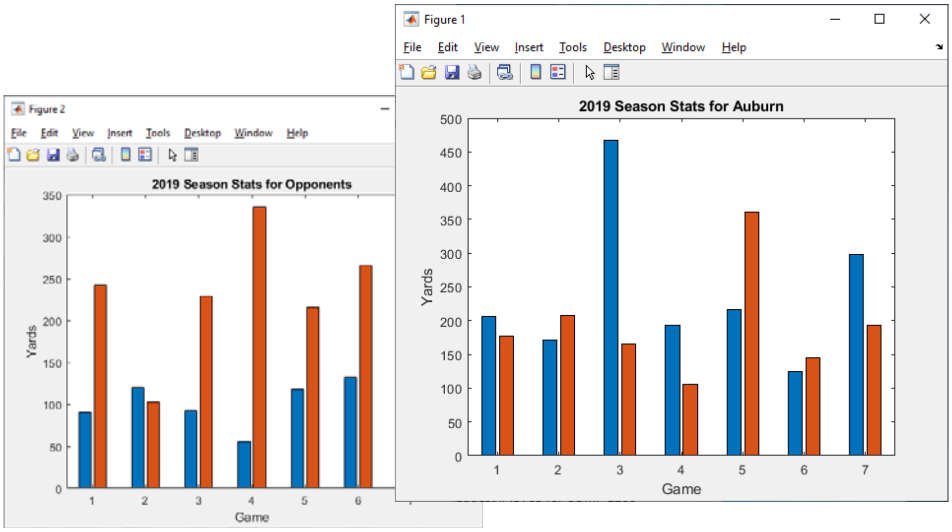


Figure 2
File Edit View Insert Tools Desktop Window Help
2019 Season Stats for Opponents
Yards
0 50 100 150 200 250 300 350
1 2 3 4 5 6
Game
Figure 1
File Edit View Insert Tools Desktop Window Help
2019 Season Stats for Auburn
Yards
0 50 100 150 200 250 300 350 400 450 500
1 2 3 4 5 6 7
Game

```
Example of textread():
   Suppose the text file mydata.dat contains data in the following form:
      Sally    Type1 12.34 45 Yes
      Joe      Type2 23.54 60 No
      Bill     Type1 34.90 12 No

   Read each column into a variable
      [names,types,x,y,answer] = textread('mydata.txt','%s%s%f%f%s');
```

Notice all rushing and passing numbers are in the rushing matrix and passing matrix.
SO, the computed averages are stored in a column of the matrix.

assign08.m will complete all requirements <u>except</u> printing the output report and drawing bar chart..
A user-defined function, `outputReportGraph()`, will do all the output.

In `assign08.m`:

Complete all requirements except the output report and graph.
Use `outputReportGraph()` to print the output report and draw graph.
To identify Auburn and Opponent report and graph, send 'Auburn' or 'Opponent' to the **`name`** and 1 or 2 to the **`figureNum`** input parameters in `outputReportGraph()`.

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

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

```
function [] = outputReportGraph( date, rushing, passing, name, figureNum )
% print stat report and draw graph
```

Instructions for all assignment scripts:

ALL script files

- ☐ 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

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

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

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

NOTE:
Your submitted file(s) MUST be spelled and cased as instructed.
[-5 points for not doing so.]

Submit via Canvas:

`assign08.m`	MATLAB script file
`outputReportGraph.m`	MATLAB function file