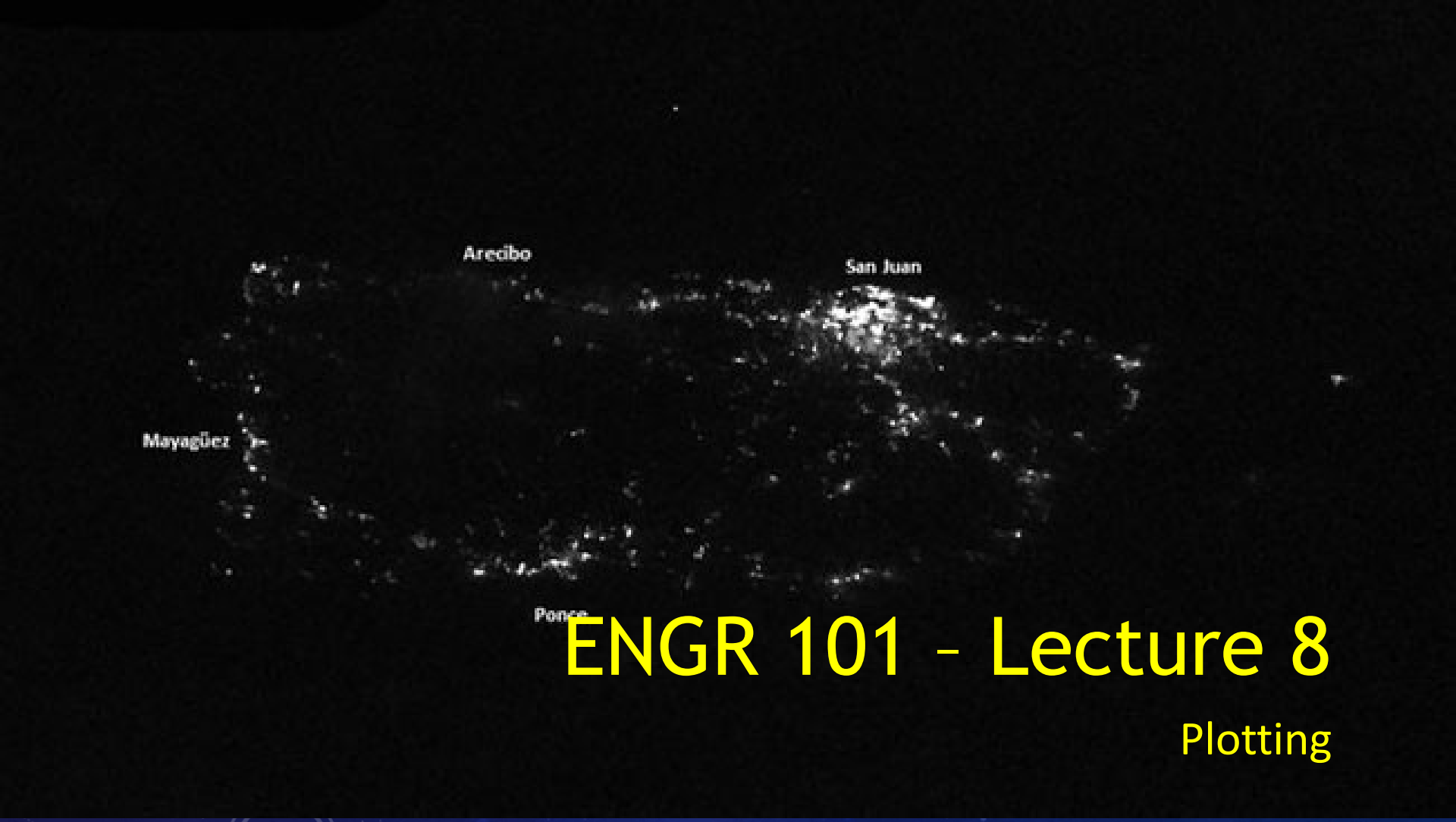
While you are waiting

Unscramble the following letters to reveal an October word

oypsko



ENGR 101 – Lecture 8

Plotting

Laura Alford, James Juett, Rick Niciejewski 10/2/17

Lecture Goals

- Previous lecture: Working with Images
  - Greyscale image representation
  - RGB image representation
  - HSV image representation
  - Suggested readings, Attaway, Chap 13.2
  - Project 2 Overview
- Todays lecture: plotting with MATLAB
  - Potpourri of MATLAB plotting commands
  - Suggested readings, Attaway, Chap 3.5, 11.1

Announcements

- I always hold “office hours” after lecture behind Stamps Auditorium.
- <u>Mondays/Wednesdays, 10:30am until done</u>
  - Project 2
  - Lecture questions
  - Any other business

Creating a Scatterplot

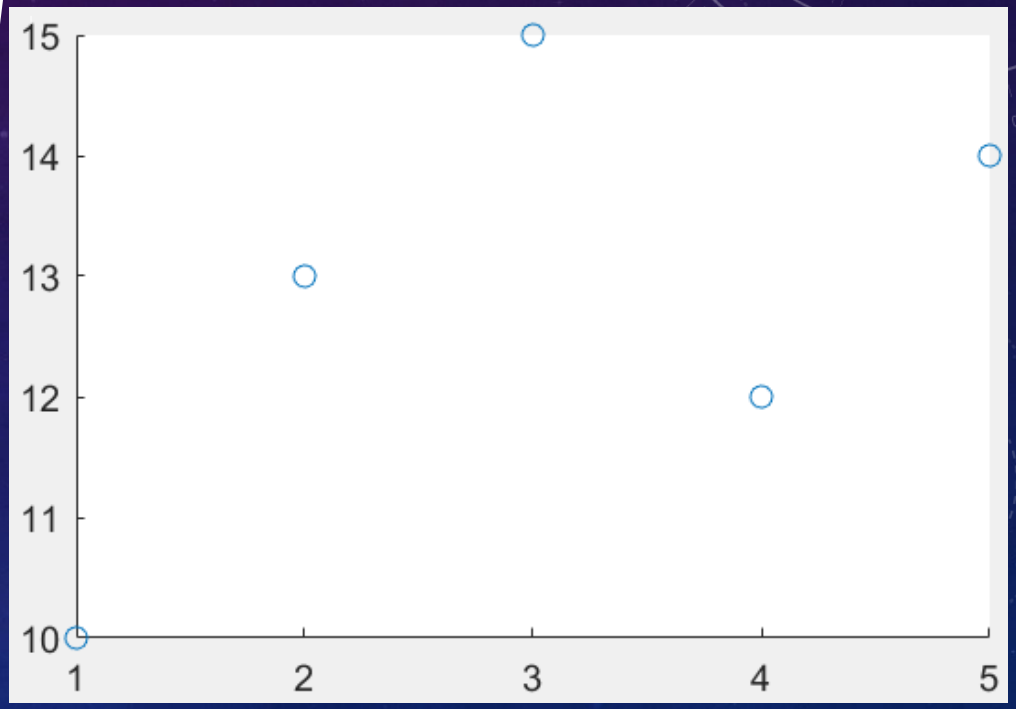
- To show a scatterplot, use the `scatter` function.

```
x = [1,2,3,4,5];
y = [10,13,15,12,14];
scatter(x,y);
```

- It takes two arguments, which are interpreted as coordinates of ordered pairs to be plotted.

(x,y)

(1,10) (2,13) (3,15) (4,12) (5,14)



The plot Function

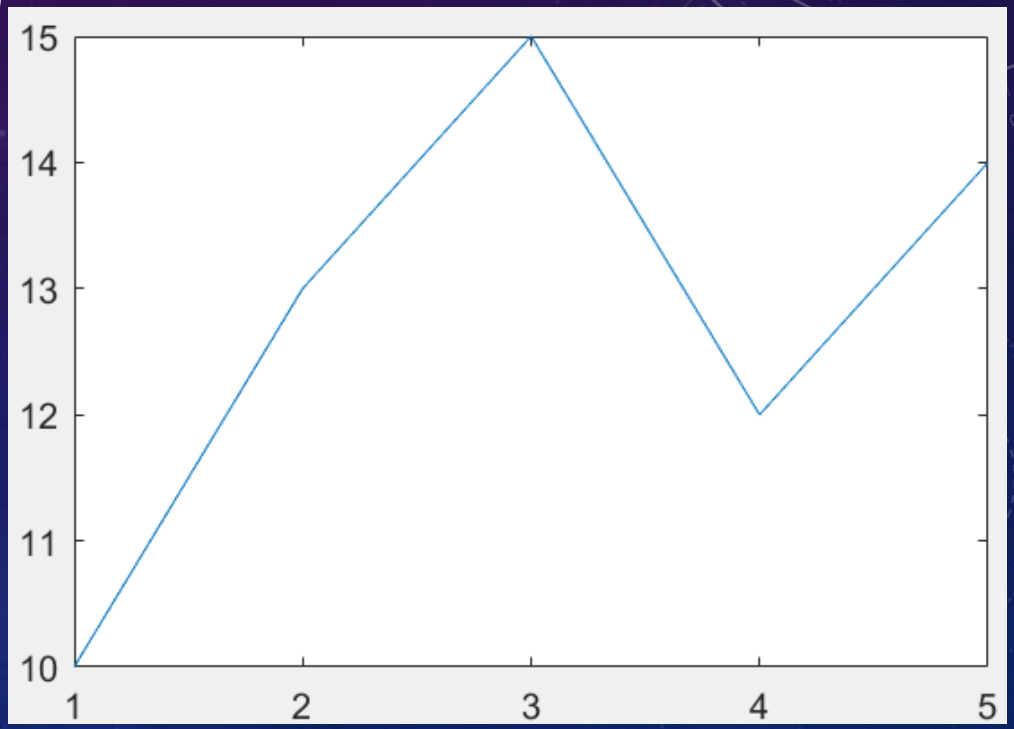
- plot works similarly to scatter, but connects the data points.

```
x = [1,2,3,4,5];
y = [10,13,15,12,14];
plot(x,y);
```

- The ordering of the (x,y) pairs matters (as opposed to scatter).

(x,y)

(1,10) (2,13) (3,15) (4,12) (5,14)

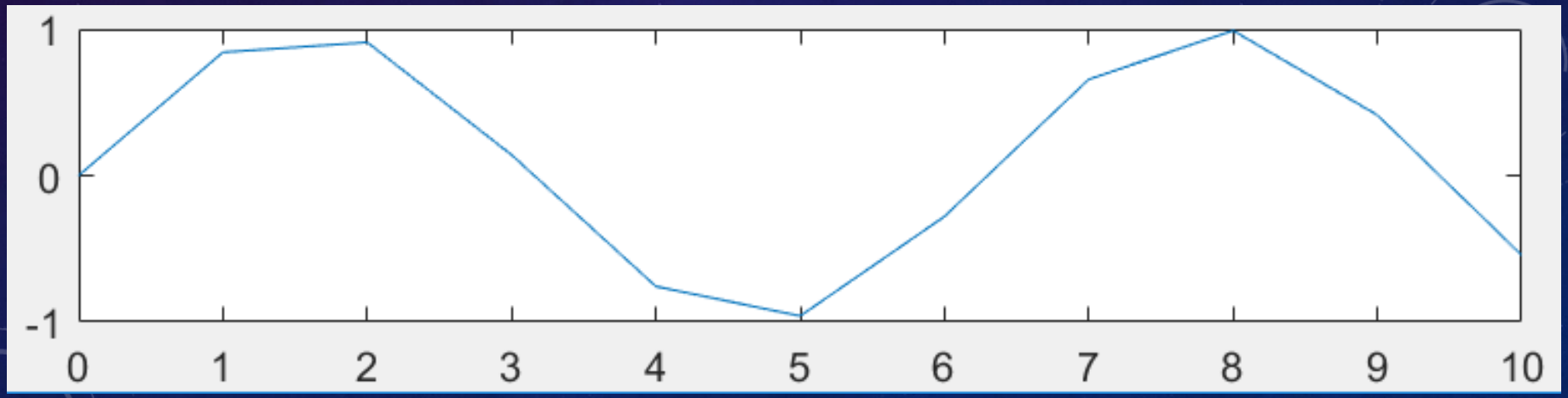


Plotting Functions

- Can we use MATLAB like a graphing calculator?
  - This doesn't work: `plot(sin);`
- `plot` requires a set of ordered pairs
  - We can generate these easily using <u>vectorized</u> code!

```
x = 0:10;
y = sin(x);
plot(x,y);
```

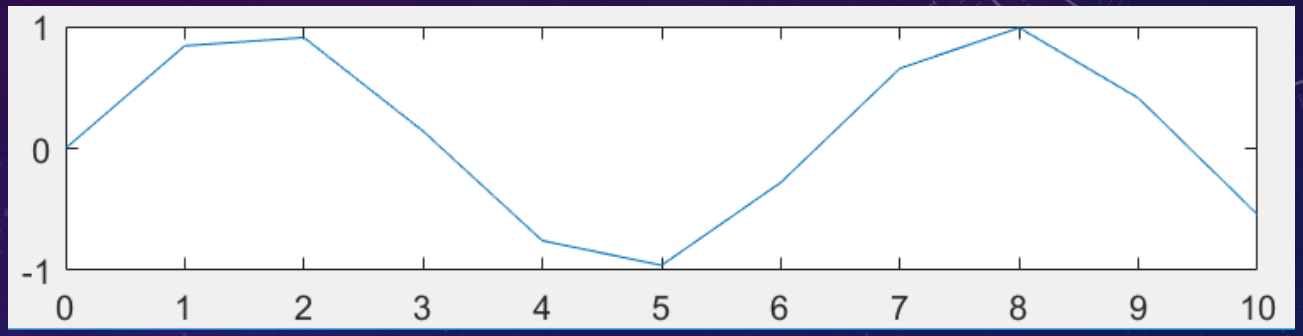
x	0	1	2	3	4	5	6	7	8	9	10
y	0	0.84	0.91	0.14	-0.76	-0.96	-0.28	0.66	0.99	0.41	-0.54



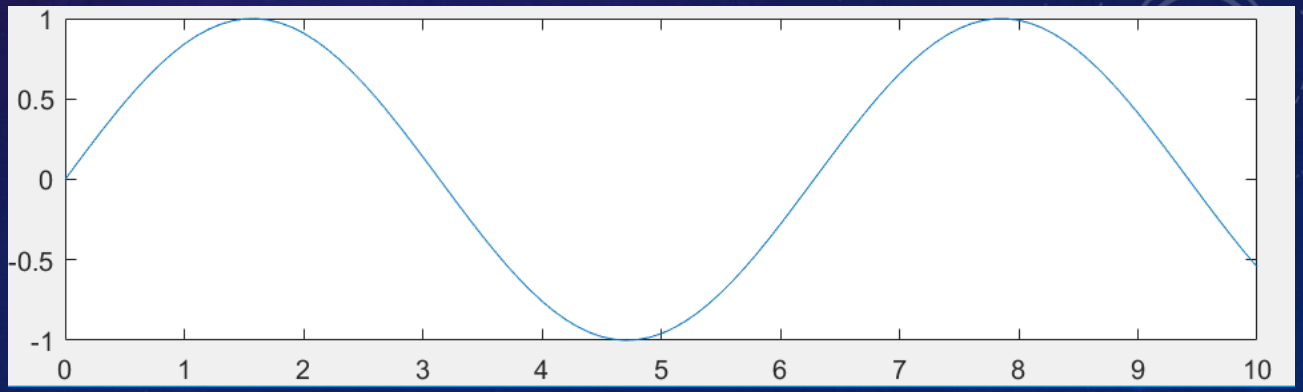
Plotting Functions

- To create a smoother plot of a math function, just use more data points. To do this with a range, decrease the step size.

```
x = 0:10;
y = sin(x);
plot(x,y);
```




```
x = 0:0.1:10;
y = sin(x);
plot(x,y);
```



Your turn #1 – projectile motion

- Write a function **`ex01`** that calculates, as a function of time, the height of a baseball thrown vertically upwards

```
function [h] = ex01(h0,vi,a,t)
```

  - Use **$h = h0 + vi*t + a*t^2/2$**

    where a = acceleration due to gravity (Google it)

  - Let time t = [0:0.1:10] s

    assume h0 = 0, vi = 40 m/s (fastball is 90 mi/hr)

  - plot(t,h) for <u>only those values with positive height</u>

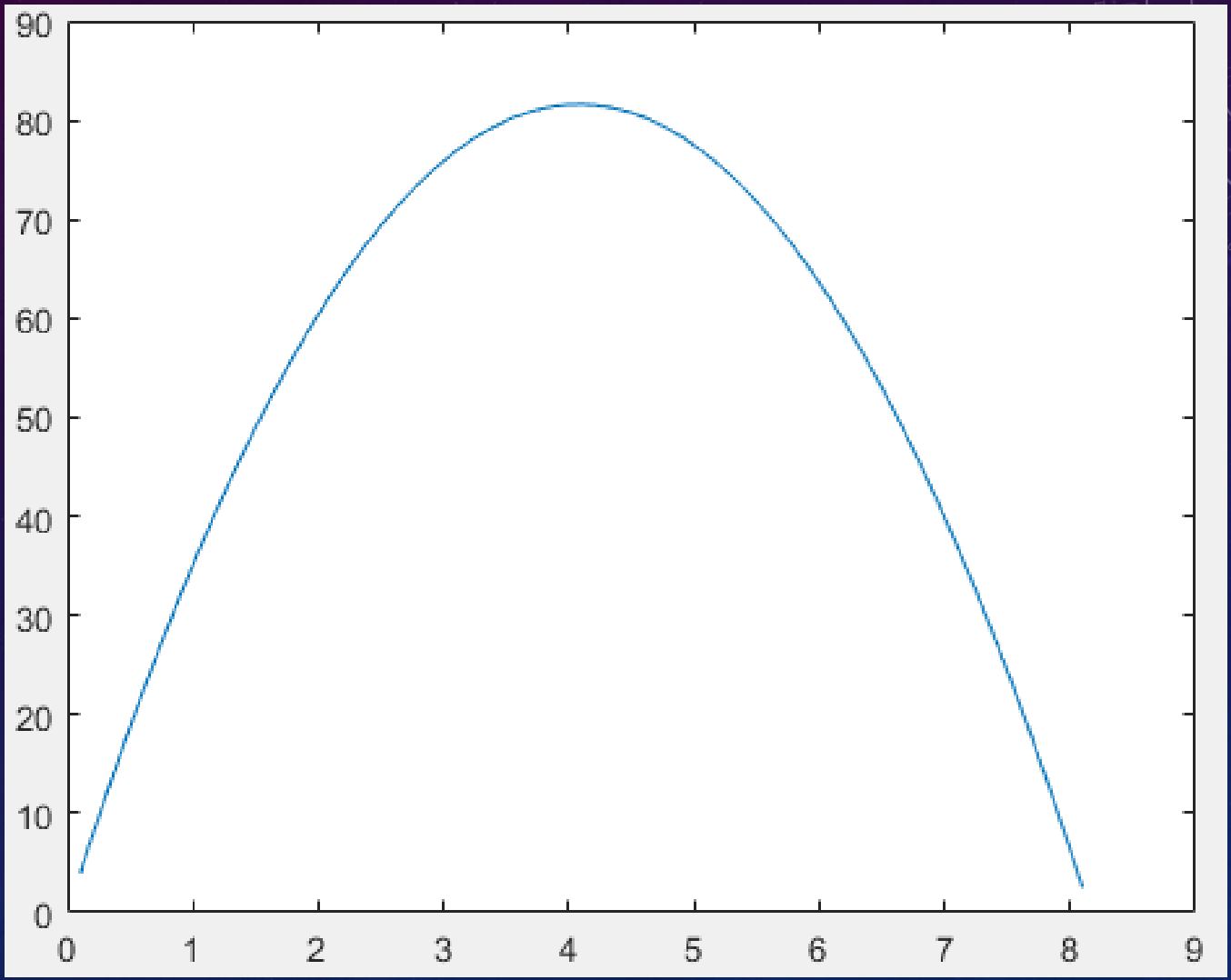
- Call your function from the command window

Exercise #1 – projectile motion

```
function [h] = ex01(h0,vi,a,t)
% vertical height, projectile motion
  h = h0 + vi.*t + 0.5.*a.*t.^2;
  plot(t(h>0),h(h>0)); % here or in a script
end
```

```
>> h=ex01(0,40,-9.8,[0:.1:10]);
```

Exercise #1 – projectile motion



Plotting Multiple Sets of Data

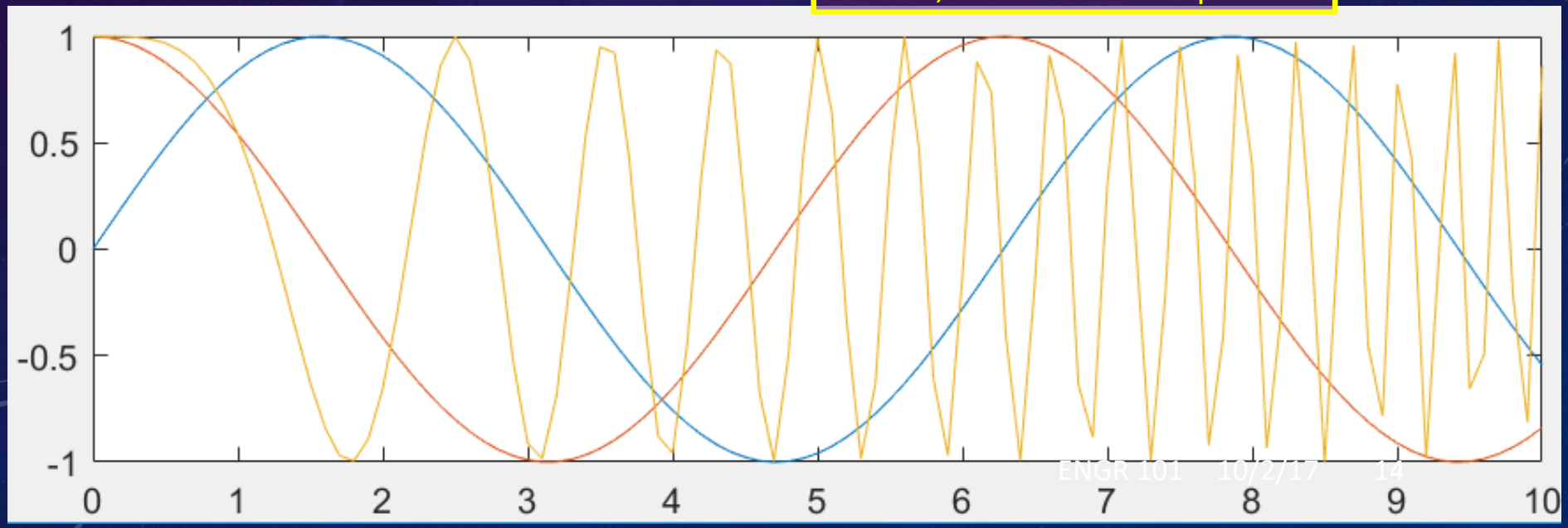
- `plot` and `scatter` allow you to specify multiple sets of data that may be shown together.

Change to `x = 0:0.01:10;`

Why does the plot of *cos(x²)* look so bad?

```
x = 0:0.1:10;
plot(x, sin(x), x, cos(x), x, cos(x.^2));
```

The "x" values will often be the same, but this is not required.



The `linspace` Function

- The linspace function provides an alternate way to create evenly spaced vectors of numbers.
- Range Notation:

```
x = [0:0.01:10];
```

Start | Step size | End

- `linspace`:

```
linspace(0,10,1001);
```

Start | End | How many numbers?

Figures

- In MATLAB, **figures** are used to display graphics (e.g. plots, charts, images, etc.) in a separate window.
- You can have several figures at once.
  - Each figure has a unique number (e.g. figure 1, figure 2, …)
  - The **current figure** will be the target of any display operations.
  - Initially, figure 1 is the current figure.
- Use the `figure` function to manage figures.
  - `figure();`
  - `figure(n);`

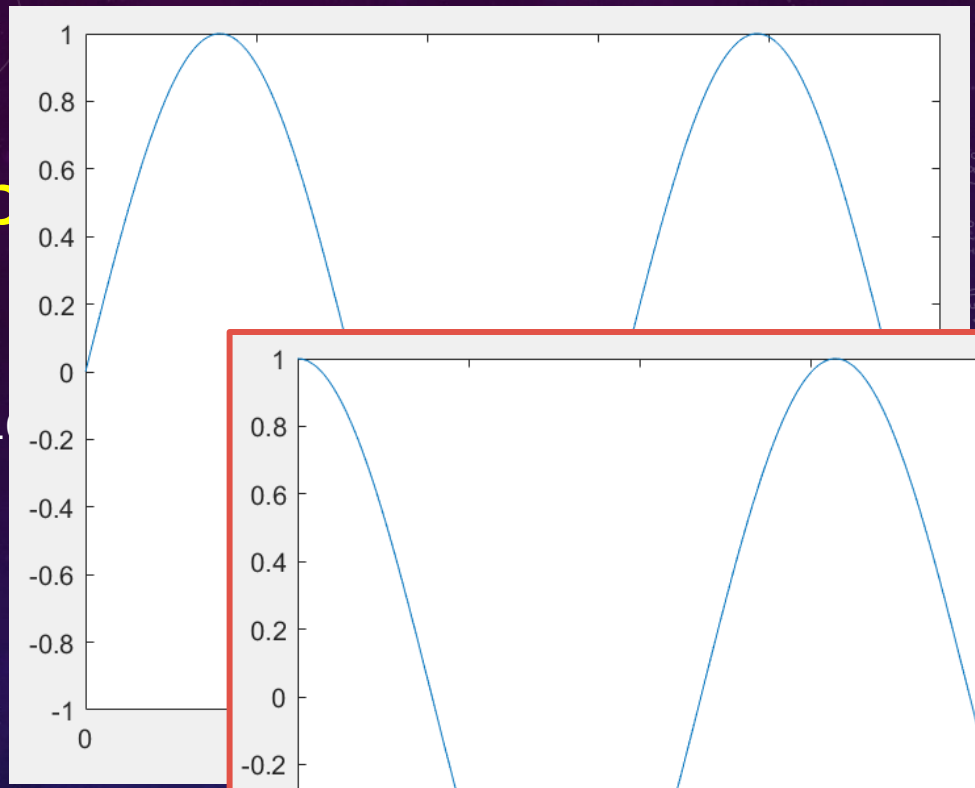
Creates a new figure, which becomes the current figure.

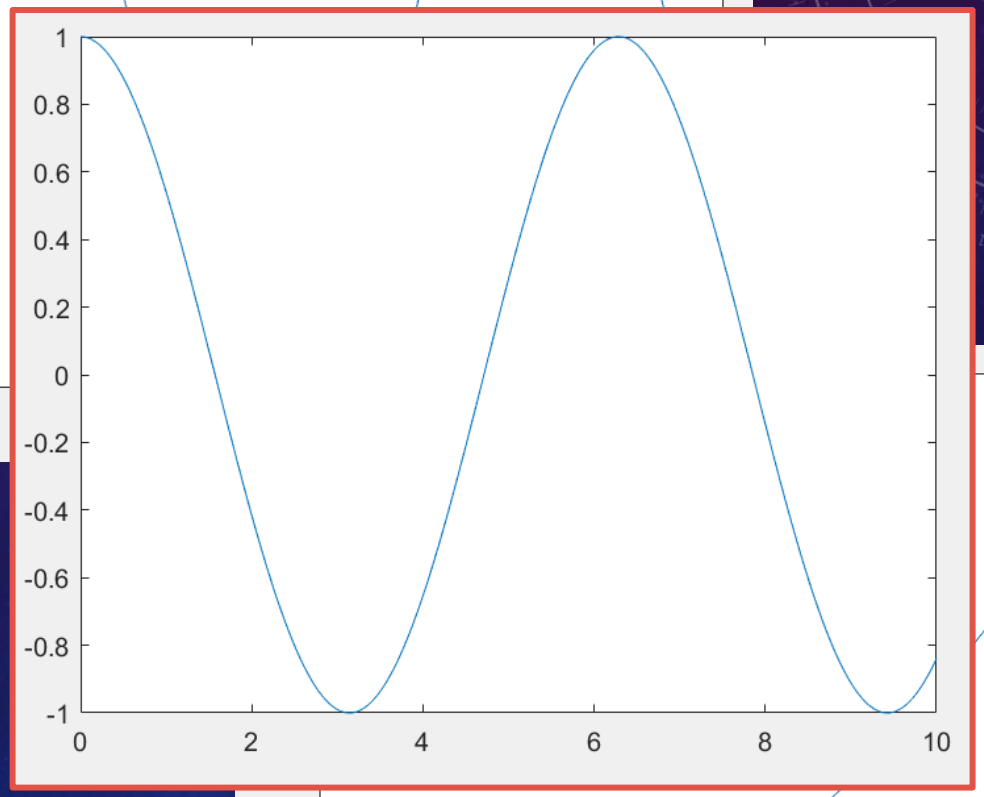
Sets figure n to be the current figure. (It is created if it doesn't exist already.)

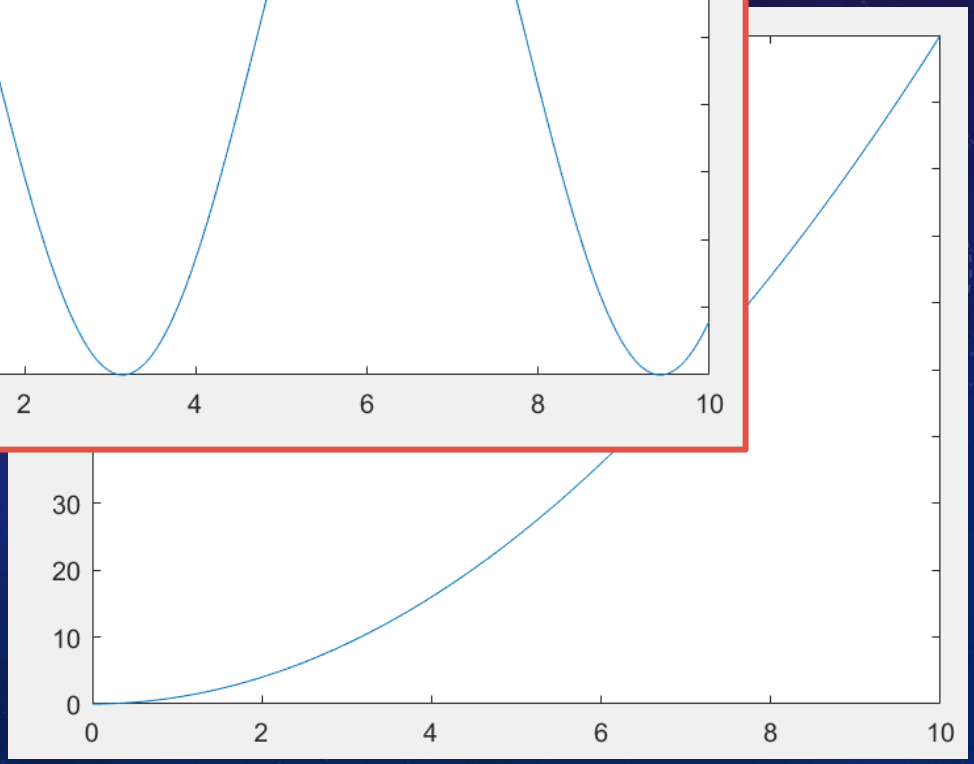
Using Multip

- Here's an example:

```
x=linspace(0,10,1

figure();
plot(x, sin(x));

figure();
plot(sin(x), x);

figure(42);
plot(x, x.^2);

figure(2);

plot(x, cos(x));
```







- The original plot in figure 2 disappeared :(

Closing Figures

- To close the current figure:

```
close
```

- To close all figures:

```
close all
```

- To close a particular figure:

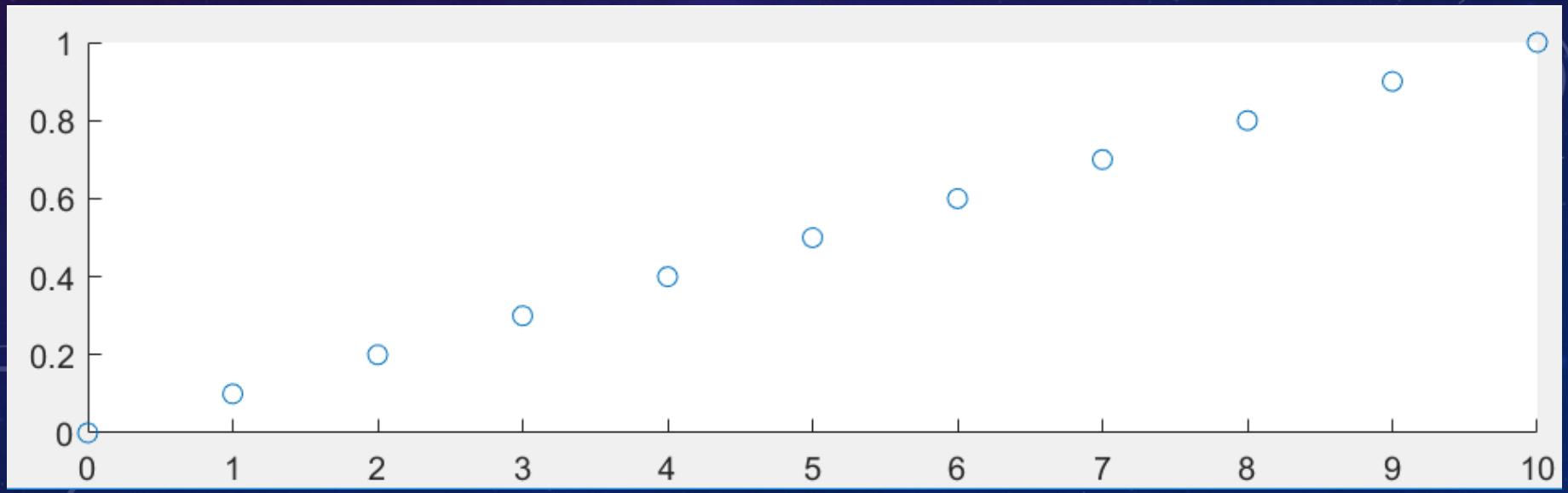
```
close n
```

Multiple Plots in One Figure

- We've seen that `plot` can accept multiple sets of data, but what if you also wanted a scatterplot on the same axes?

```
figure();

x1 = 0:0.1:10;
plot(x1, sin(x1));

x2 = 0:10;
scatter(x2, x2 ./ 10);
```

MATLAB's default behavior is to replace the old plot with the new one.

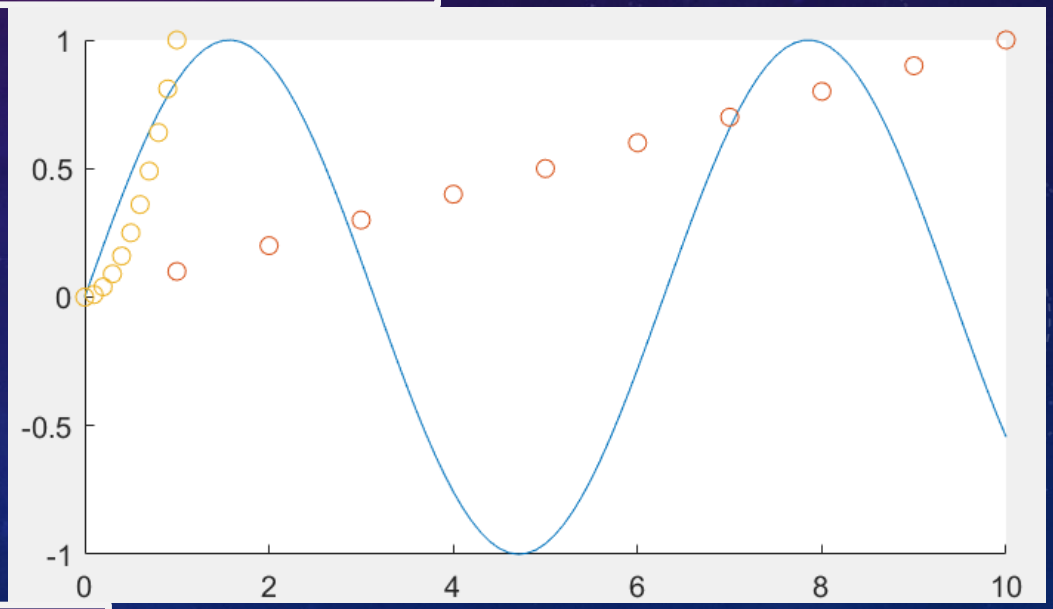


The hold Command

- The hold command tells MATLAB to add new plots to the same set of axes instead of replacing the old plot.

```
figure();
hold on;
x1 = 0:0.1:10;
plot(x1, sin(x1));
x2 = 0:10;
scatter(x2, x2 ./ 10);
x3 = 0:0.1:1;
scatter(x3, x3 .^ 2);
hold off;
```

MATLAB will now add new plots without replacing the old ones.



After hold is turned off, new plots will once again replace old ones.

Your turn #2 – actual projectile motion

- Same as before, but the ball is thrown at a 45 deg angle. Write a new function that determines the (x,y) positions of the ball over the same time period, t, starting at x0=y0=0, vi=40, t=[0:0.1:10].
  - x = x0 + vi*t*cos(theta)
  - y = y0 + vi*t*sin(theta) + a*t^2/2

```
function [x,y] = ex02(x0,y0,vi,a,t,theta)
```

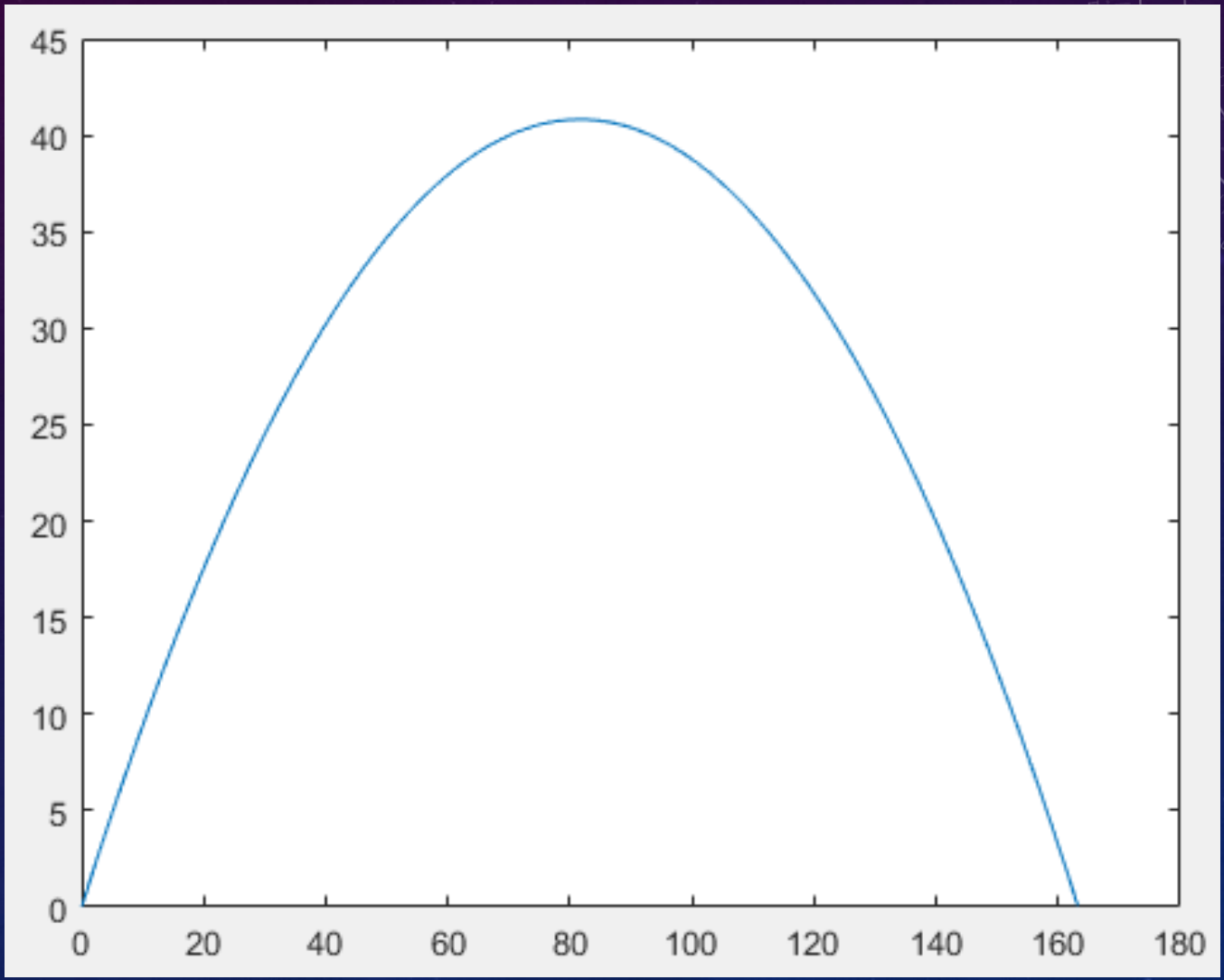
- The function must have two returns, vectors [x, y], which are both a function of time, t. Change h0 to x0,y0, and add theta as parameter inputs
- Call your function from the command window and plot(x,y) for only those values with positive height (y)

Exercise #2 – actual projectile motion

```
function [x,y] = ex02(x0,y0,vi,a,t,theta)
% vector heights, projectile motion
  ang = theta.*pi./180.;
  x = x0 + vi.*t.*cos(ang);
  y = y0 + vi.*t.*sin(ang) + 0.5.*a.*t.^2;
  plot(x(y>=0),y(y>=0))
end
```

```
>> [x,y] = ex02(0,0,40,-9.8,[0:0.1:10],45);
```

Exercise #2 – actual projectile motion



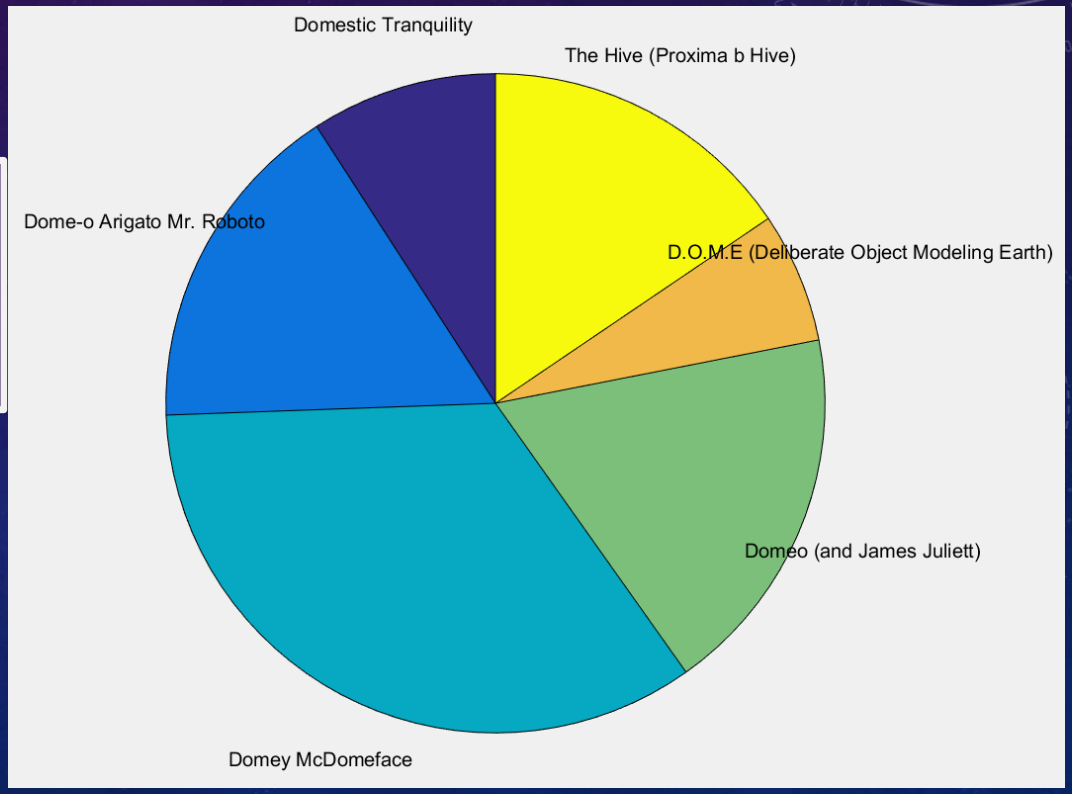
Creating a Pie Chart

- To show a pie chart, use the `pie` function.

```
votes = [20,36,75,40,14,34];
names = {'Domestic Tranquility', 'Dome-o Arigato
Mr. Roboto','Domey McDomeface','a','b','c'};
pie(votes, names);
```

The first input is a vector of either counts or percentages.

The next input contains the labels for each category.



- The `pie3` function works similarly but produces a 3D pie chart instead.

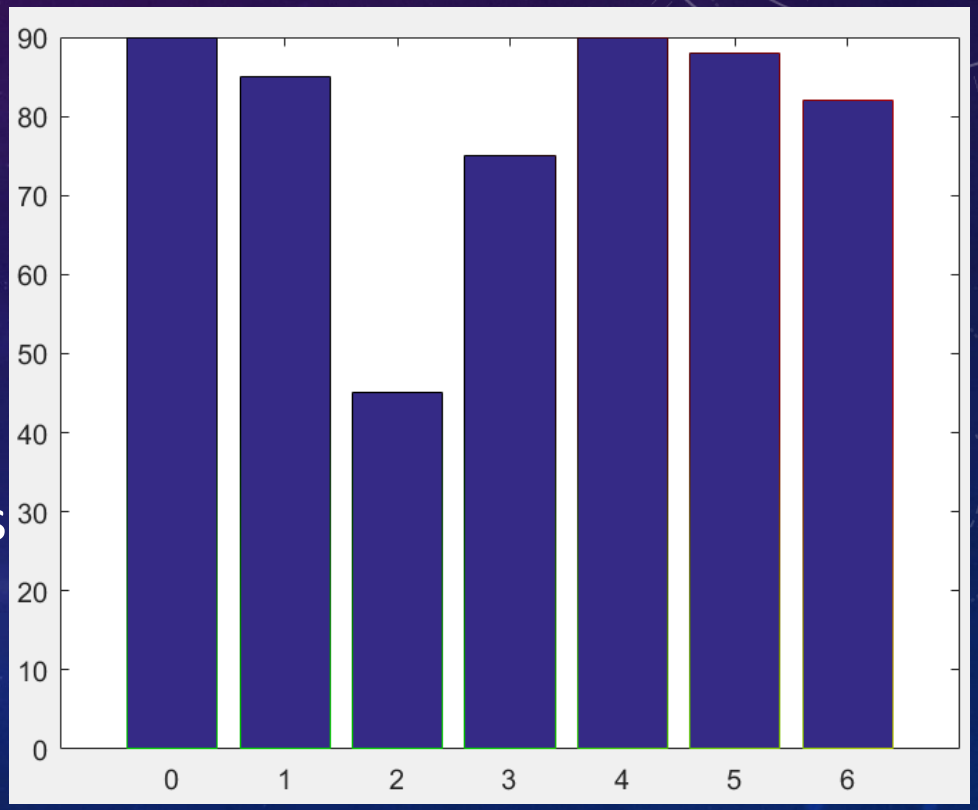
Creating a Bar Chart

- To show a bar chart, use the bar function.

```
projects = [0,1,2,3,4,5,6];
scores = [90,85,45,75,90,88,82];
bar(projects, scores);
```

The first input controls default labels and bar positions.

The next input contains the amount for each bar.



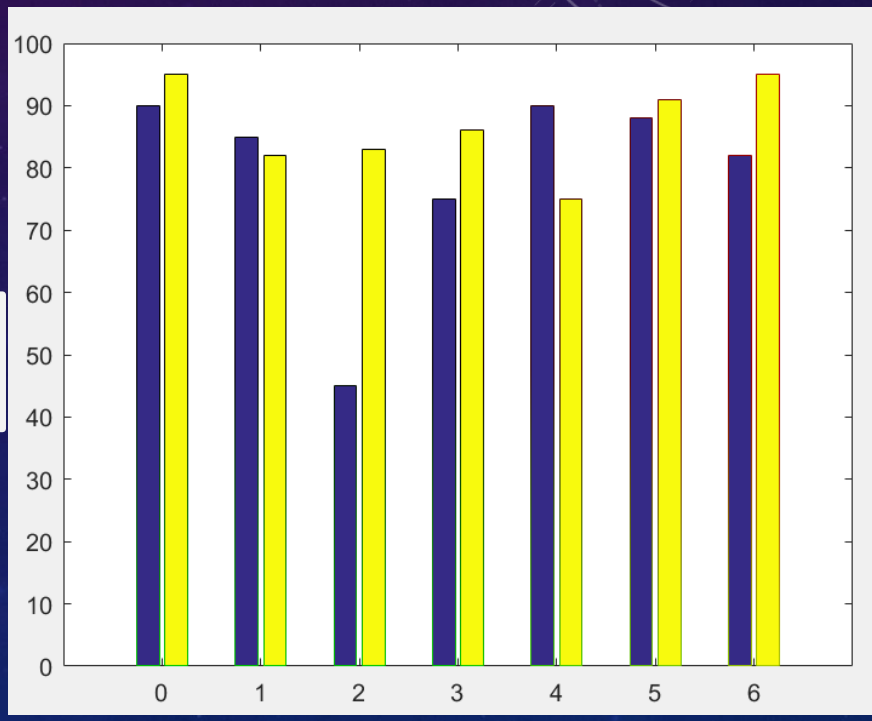
- The barh and bar3 functions work similarly but produce horizontal and 3D charts, respectively.

Creating a Bar Chart

- To show grouped bars, use a matrix where each row corresponds to a single group of bars.

```
projects = [0,1,2,3,4,5,6];
autograder = [90;85;45;75;90;88;82];
style = [95;82;83;86;75;91;95];
bar(projects, [autograder,style]);
```

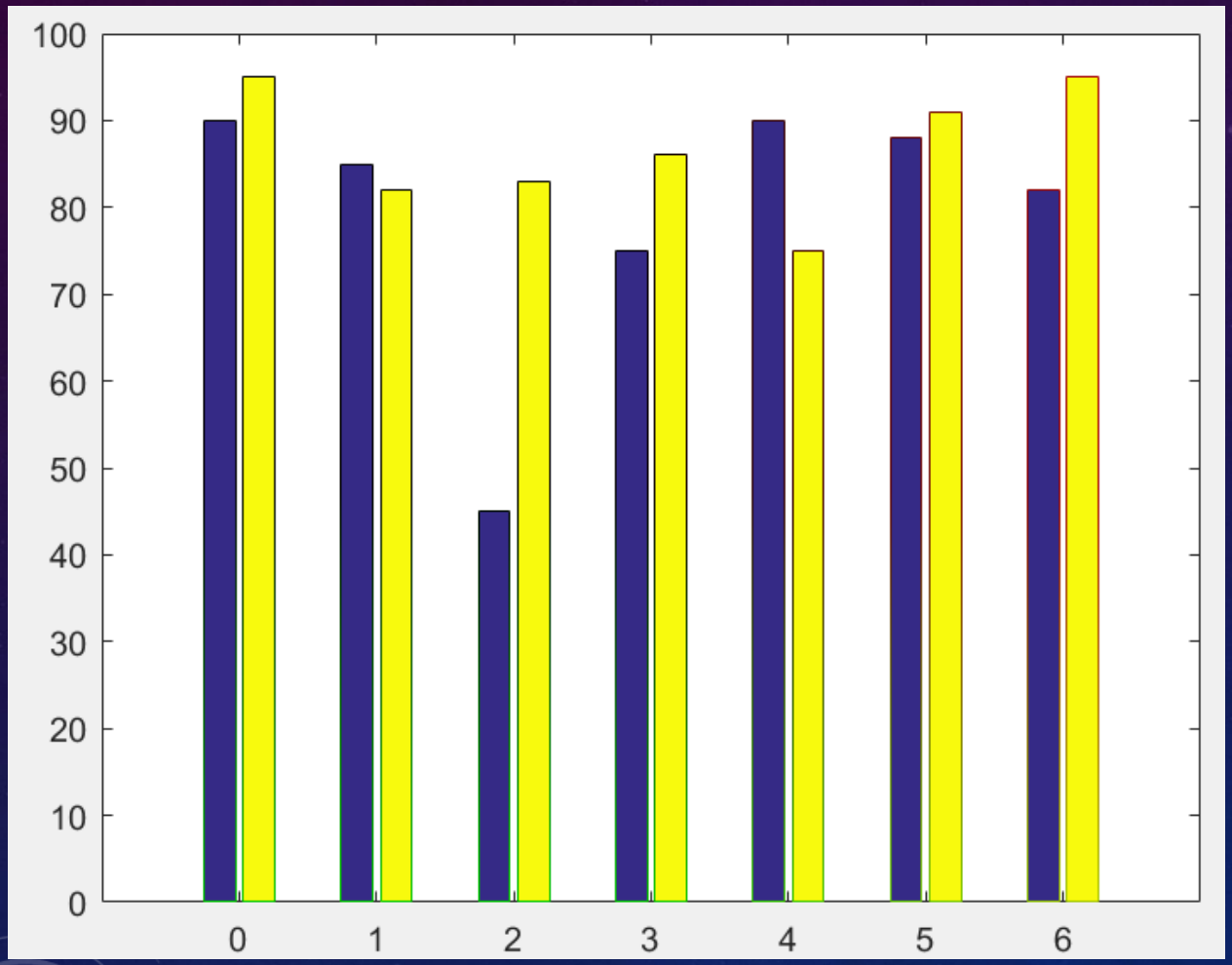
Combine so that each row has one score for both autograder and style.



Break Time

We'll start again in 5 minutes.

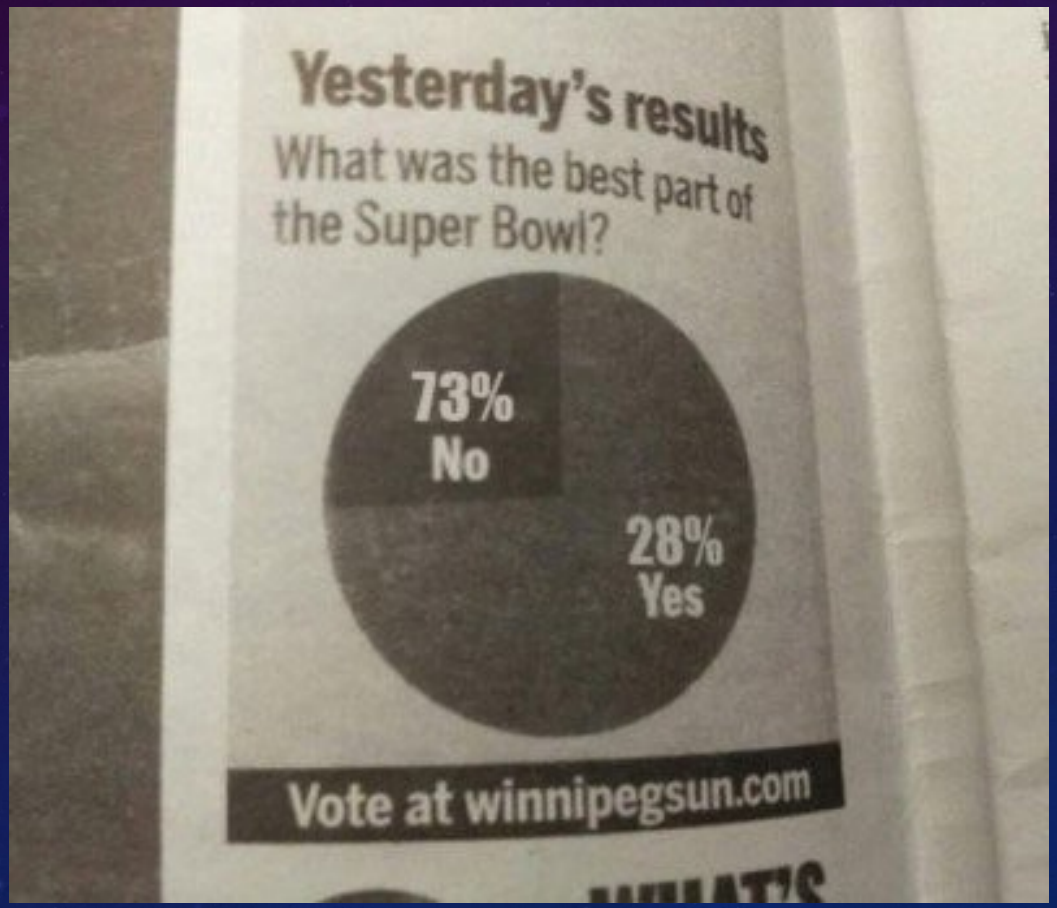
What's wrong with this chart?



How is anyone going to know what this chart is supposed to be?

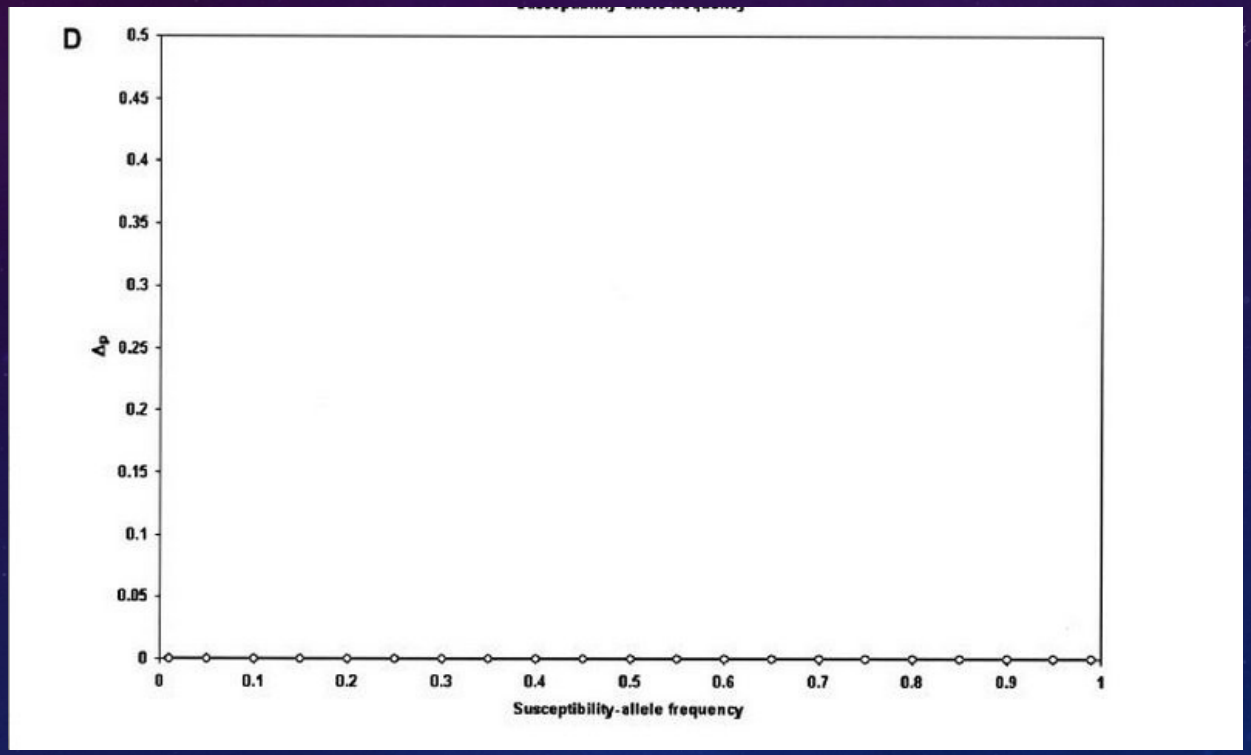
Motivation for Making Good Plots

- Because someone did this



Motivation for Making Good Plots

- Because someone did this, and this

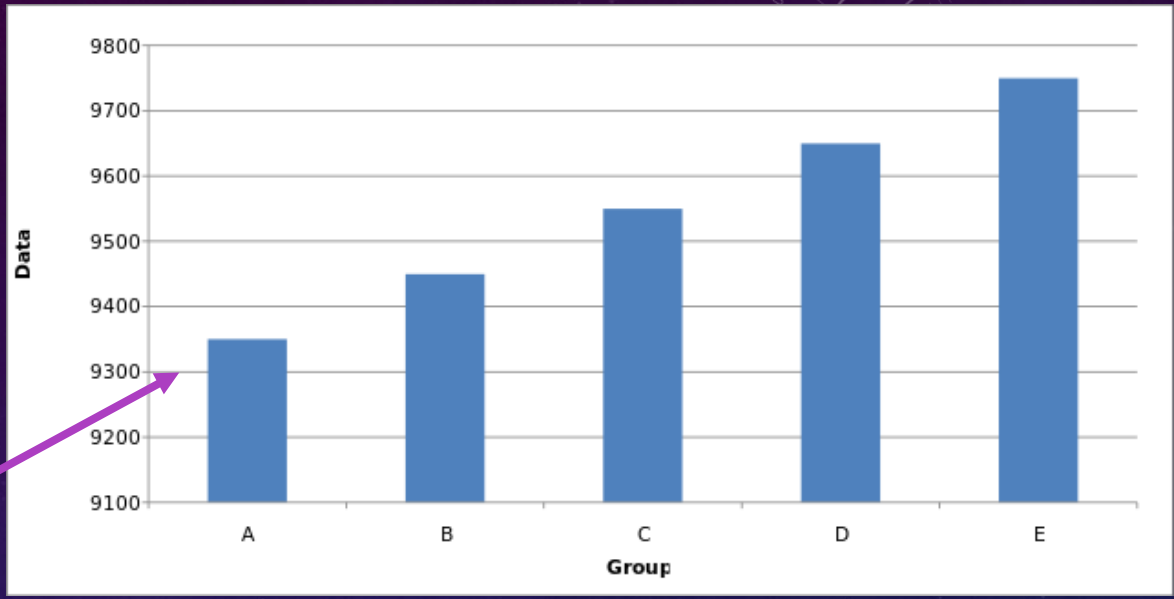


Wittke-Thompson JK, Pluzhnikov A, Cox NJ (2005) Rational inferences about departures from Hardy-Weinberg equilibrium. *American Journal of Human Genetics* 76:967-986

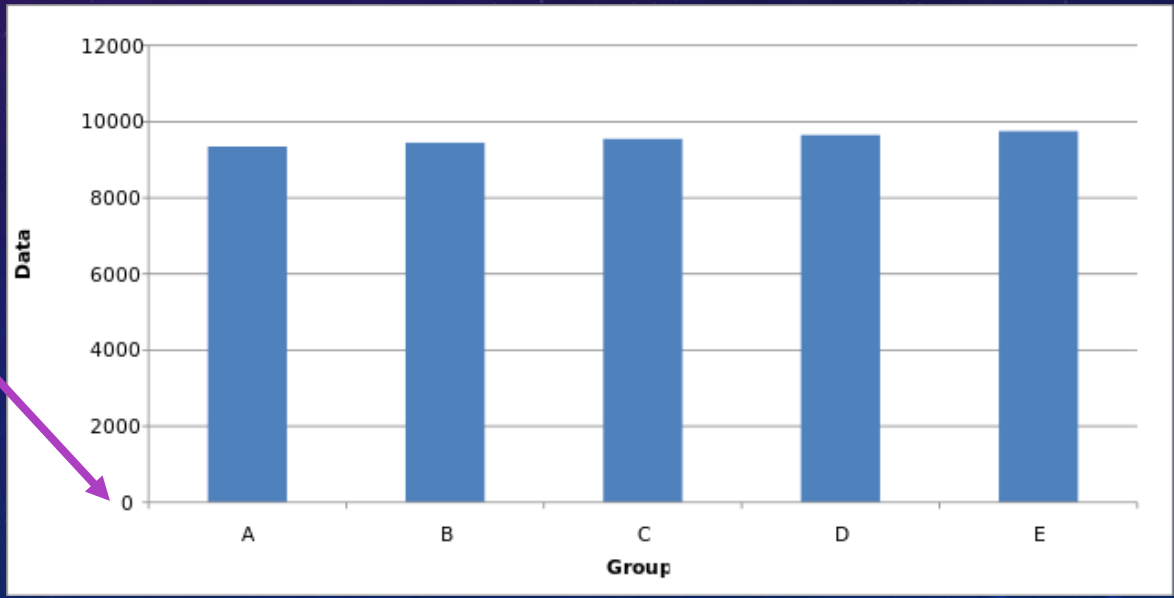
Your Graphs Will Out-Live You

- The graphs and figures you make will go on to have a life of their own
- **To be successful in engineering or science, create figures that will "stand on their own"**
- Graphs and figures can show relationships between data that are difficult, if not impossible, to explain with words
- Graphs and figures can help you (and your audience) grasp what an equation actually *means*

Don't Mislead with Your Plots



Wow! Look at the differences between the groups!



Oh wait… When the y-axis starts at zero, there's almost no difference.

for more information: https://en.wikipedia.org/wiki/Misleading_graph

Do Maximize your "Signal-Noise Ratio"

remove this

delete the title
(*if you MUST have a title, describe the relationship between x,y*)

Use a white background

add vertical grid lines

Consider colour blindness!

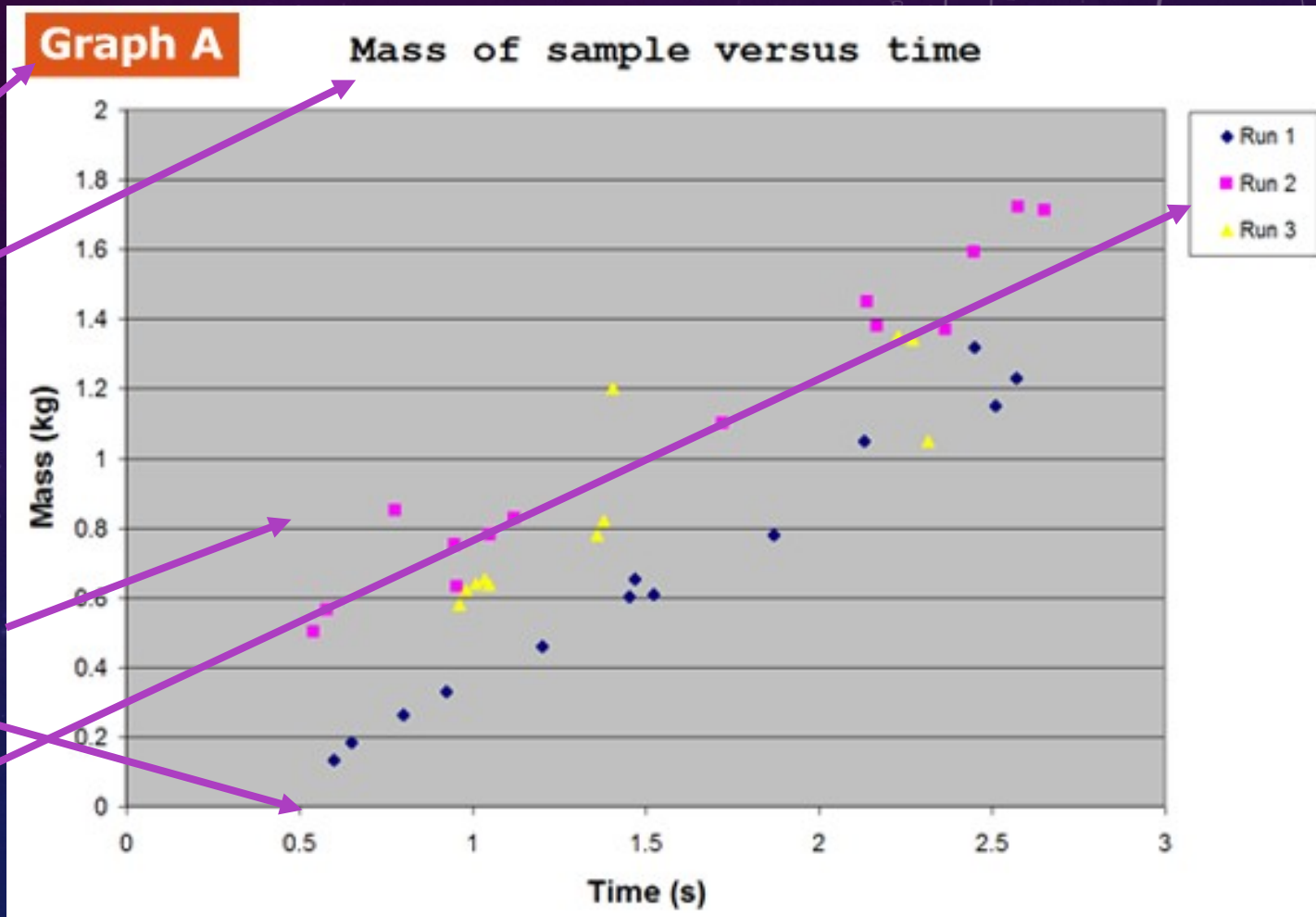


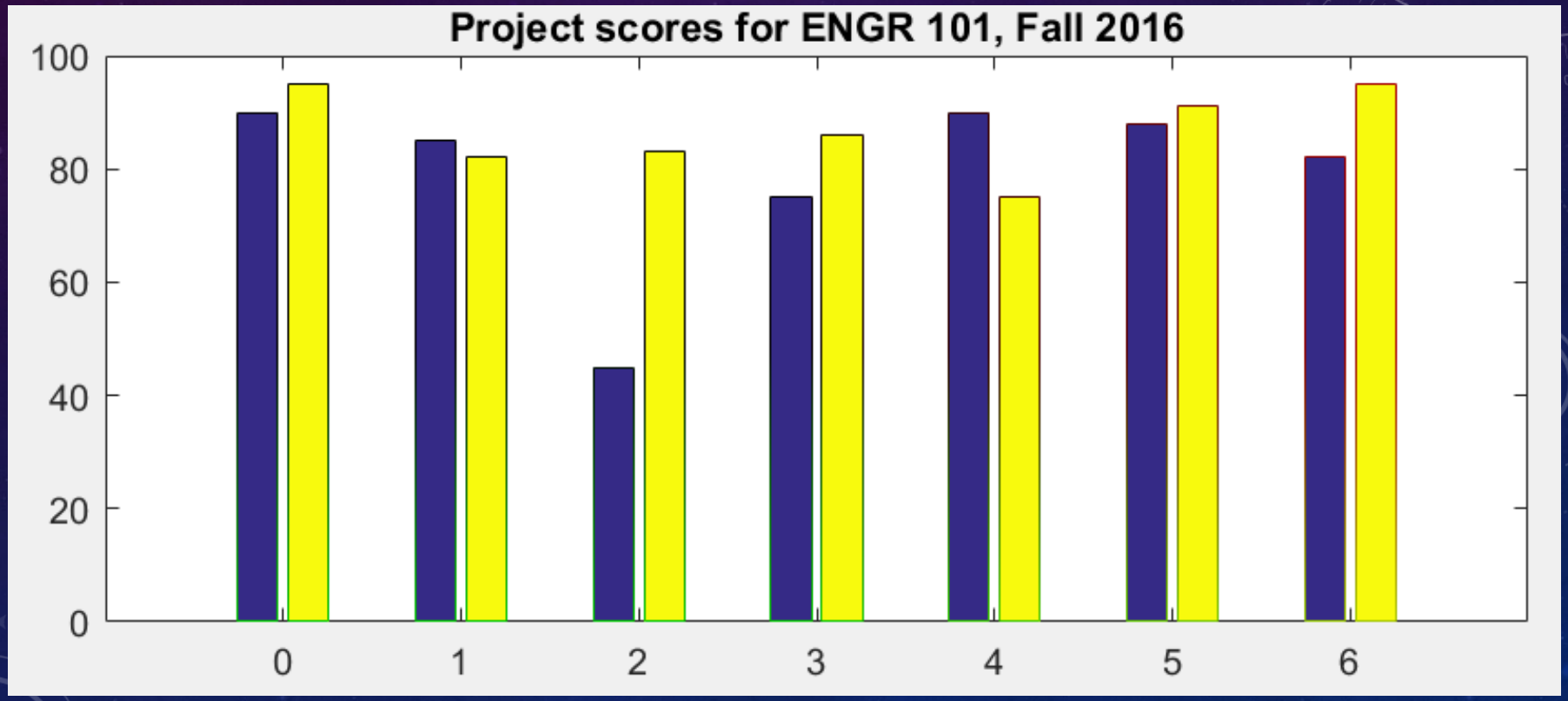
Figure 6 : Sample mass as a function of time for the three experiments.

The caption must stand on it's own: mass of ?, time since ?, runs are ?

Adding a Title

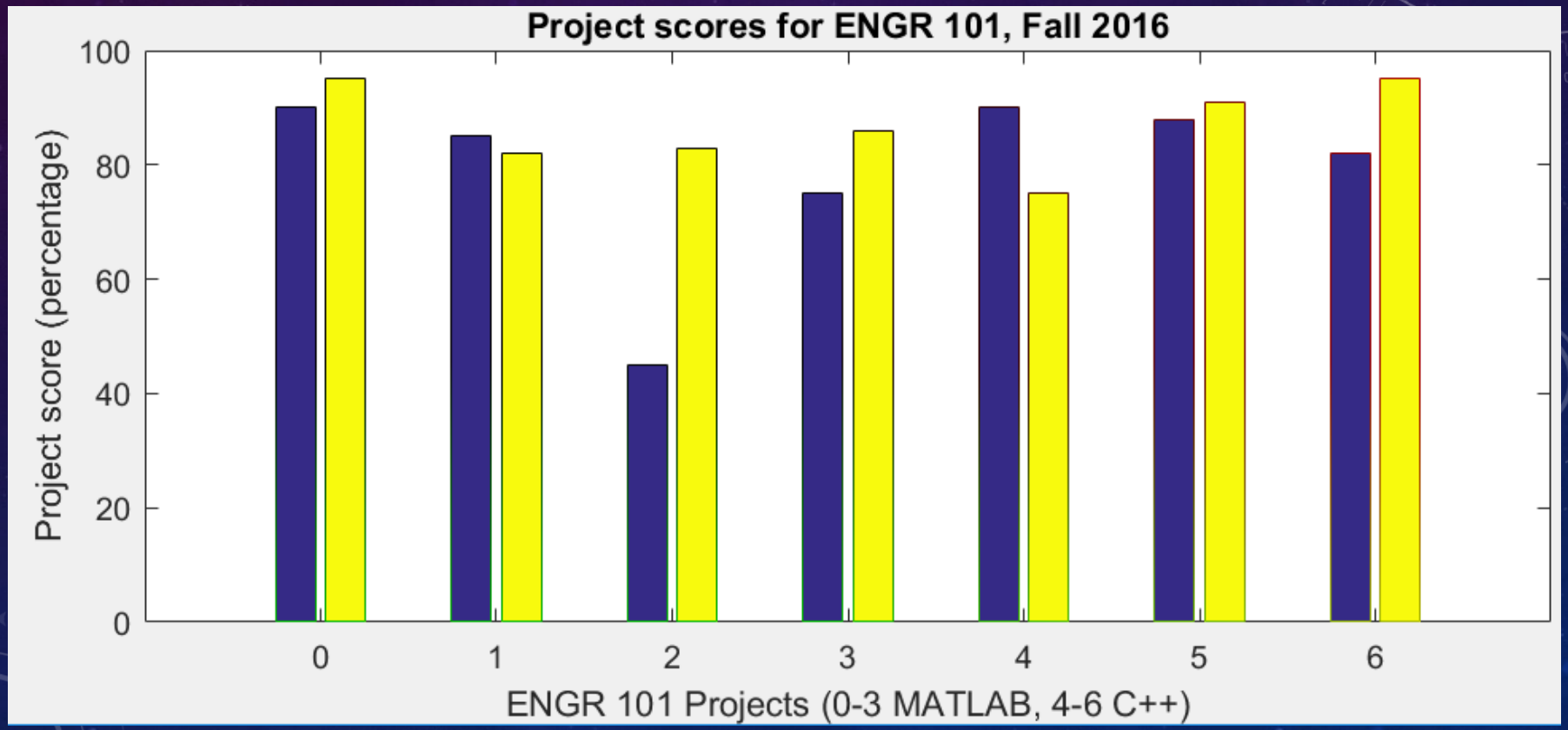
- The `title` function adds a title to the current figure.

```
title('Project scores for ENGR 101, Fall 2016');
```



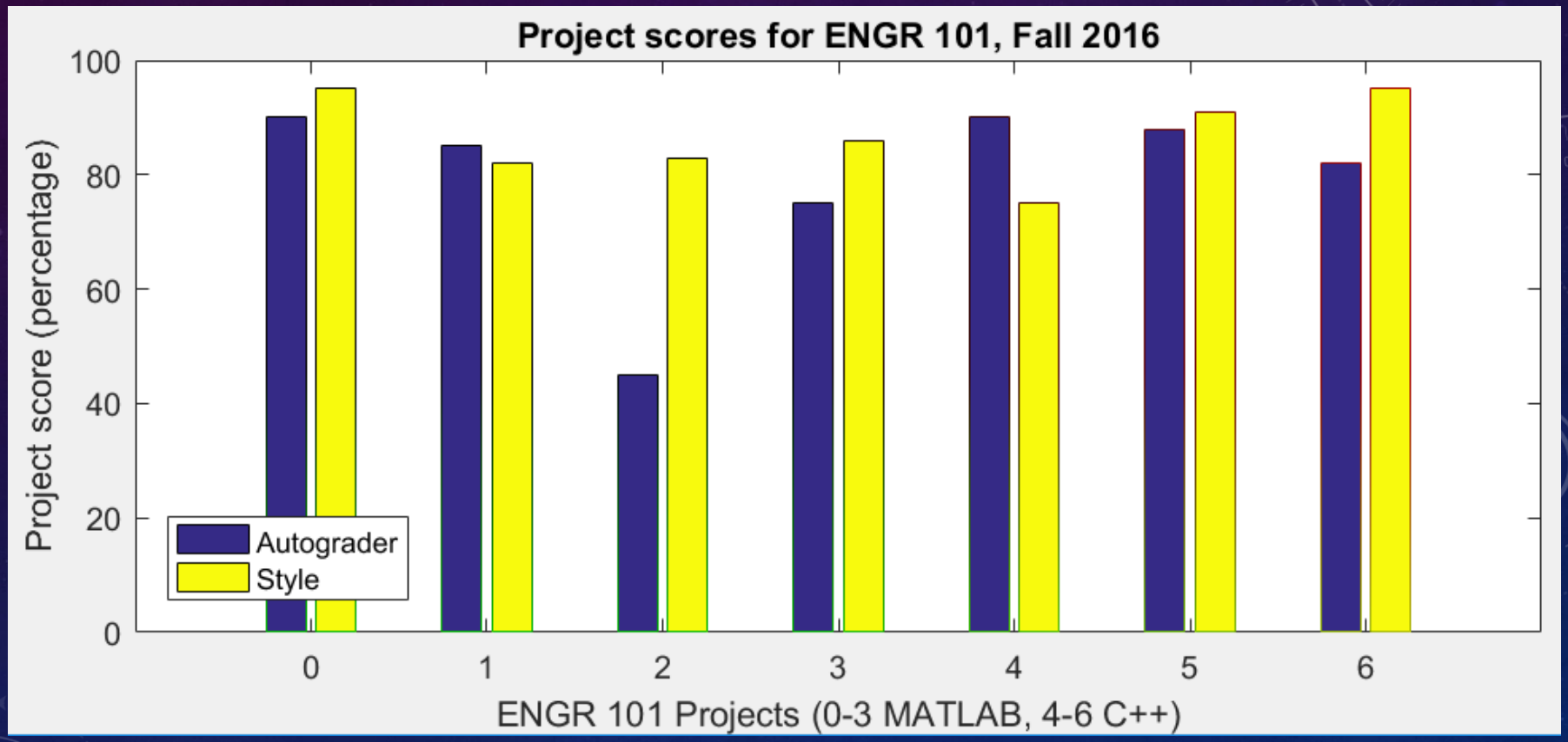
Adding Axis Labels

```
xlabel('ENGR 101 Projects (0-3 MATLAB, 4-6 C++)');
ylabel('Project score (percentage)');
```



Adding a Legend

```
legend('Autograder', 'Style');
```



Plotting and Scripts/Functions

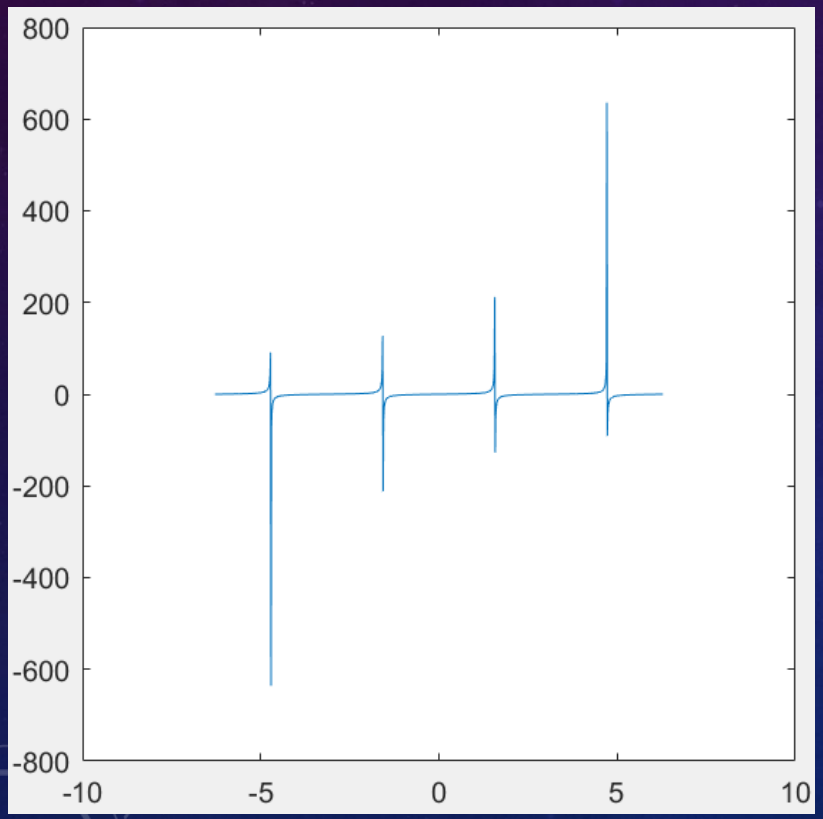
- Scripts and/or functions can be helpful as a way to collect together all the individual commands to customize a plot.
- You can change just a bit and then rerun the whole thing.

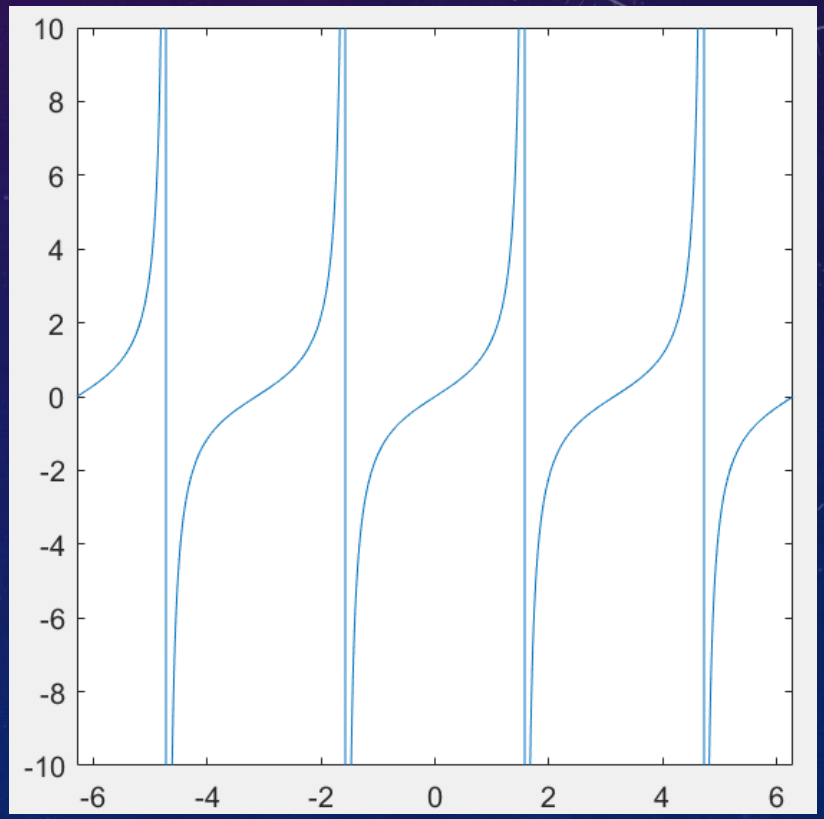
```
function [ ] = plot101Grades(projects, autograder, style)

  figure();
  bar(projects, [autograder,style]);
  title('Project scores for ENGR 101, Fall 2016');
  xlabel('ENGR 101 Projects (0-3 MATLAB, 4-6 C++)');
  ylabel('Project score (percentage)');
  legend('Autograder', 'Style');

end
```

Customizing Axes

- MATLAB tries to guess appropriate axes ranges, but sometimes it just takes a human touch. Try plotting the tan function !!!

```
xlim([-2.*pi, 2.*pi]);
ylim([-10, 10]);
```





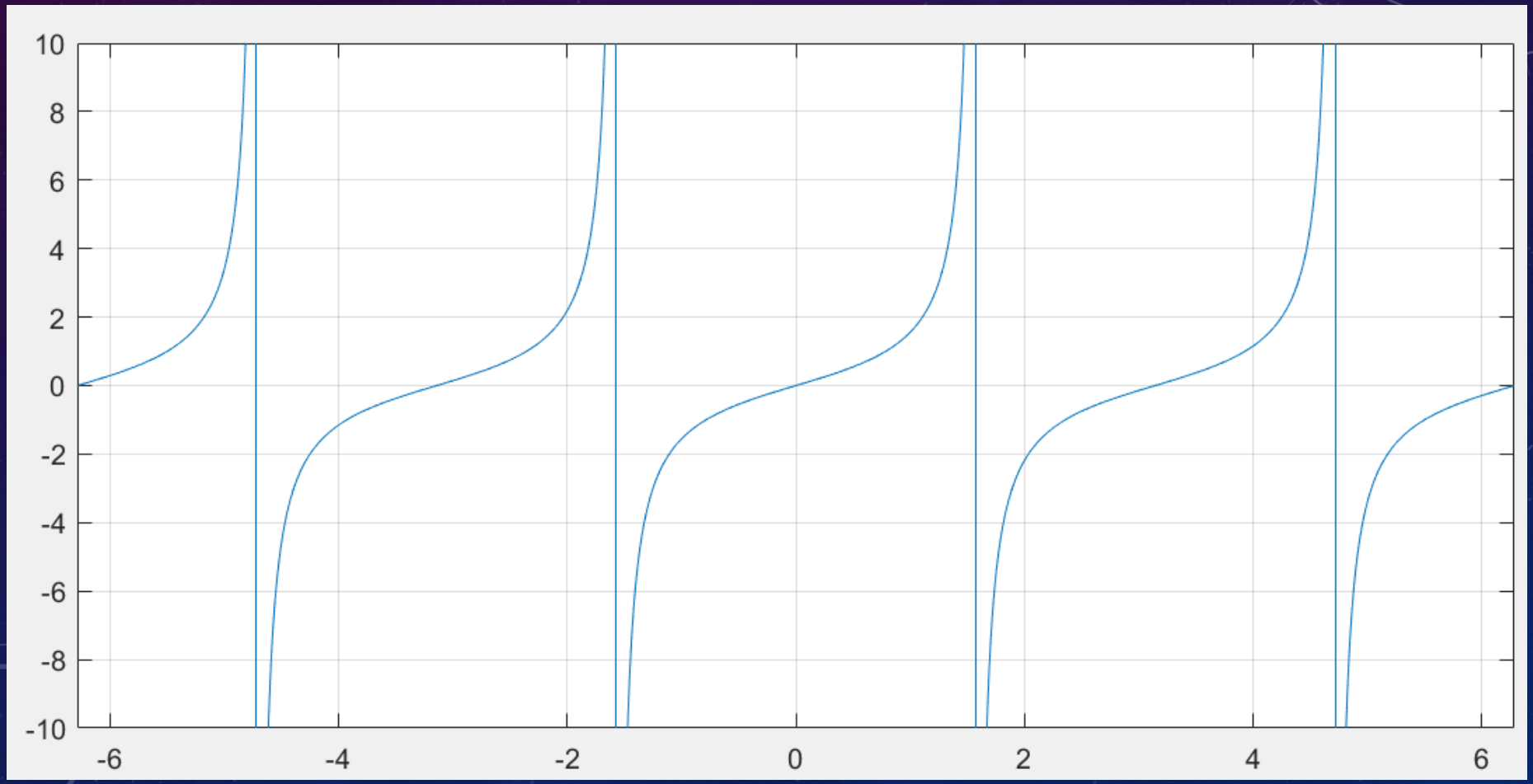
Gridlines

- To turn on gridlines:

```
grid on
```

- To turn off gridlines:

```
grid off
```



Customizing Line Plots

```
x = 0:0.1:10;
plot(x, sin(x), '--sg', x, cos(x), ':or');
```

o
circle
marker

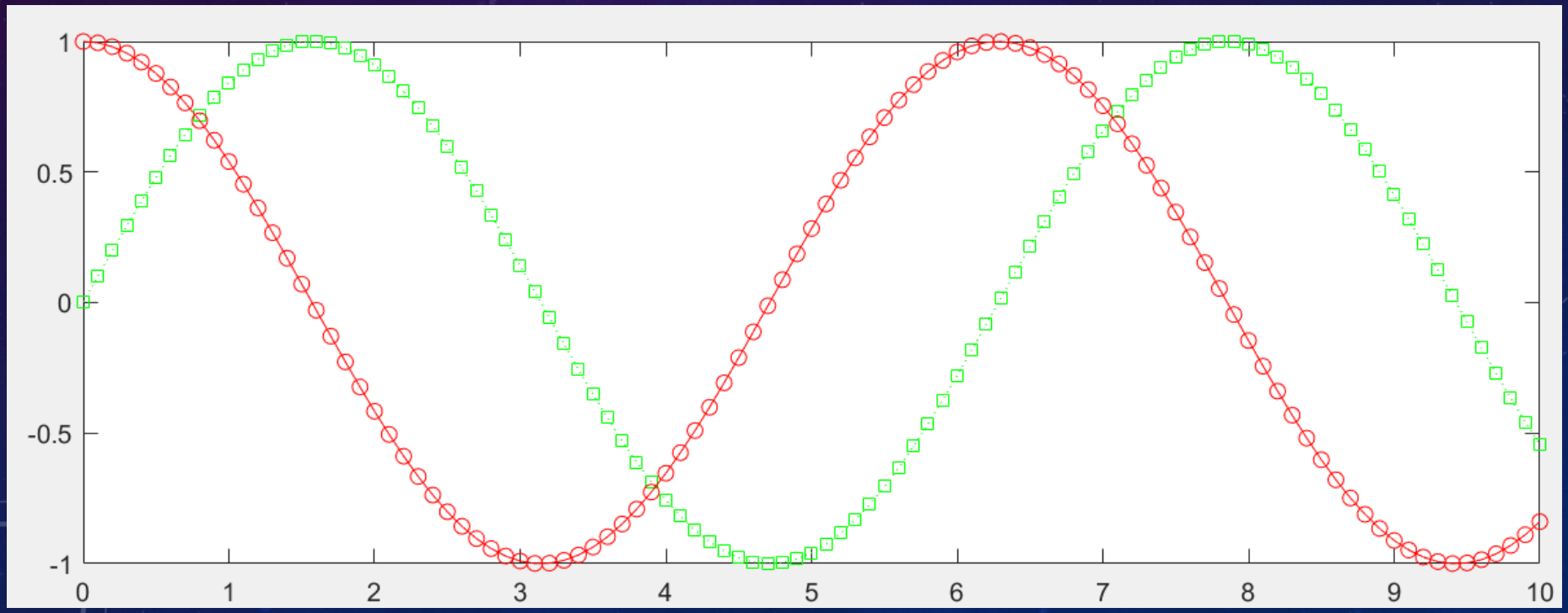
r
red color

--
dashed line

s
square
marker

g
green color

:
dotted line



Plotting in General

- So far, we've only scratched the surface of what you can do with plots in MATLAB. There's so many more options!
- Here's the truth:

Nobody memorizes all the different kinds of plots and the ways you can customize them.

- Refer to online documentation for general guidance.
- Search online if there's something specific you're looking for.
  - https://www.mathworks.com/help/matlab/creating_plots/types-of-matlab-plots.html

5 min

Challenge: Rainfall Data

- Download the file `rainfall.mat` from the Google Drive.
- Use `load('rainfall.mat')` to load the dataset:
  - days – Numbered days 1-276 (October 2nd is day 276)
  - dailyRain – Amounts of rainfall on each day.
  - totalRain – A cumulative sum of rainfall up to that day.
- Create charts that show both the daily rainfall amounts and the cumulative total as a function of the day.
  - Think about which kind of plot is appropriate to use for each case!

http://www3.a2gov.org/RainGauges/

See you Wednesday