In [62]:
```
# Initialize OK
from client.api.notebook import Notebook
ok = Notebook('hw1.ok')
```

```
=====================================================================
Assignment: hw1
OK, version v1.13.11
=====================================================================
```

HW 1: Math Review and Plotting

Due Date: Monday 1/28, 11:59 PM

Collaboration Policy

Data science is a collaborative activity. While you may talk with others about the homework, we ask that you **write your solutions individually**. If you do discuss the assignments with others please **include their names** at the top of your notebook.

Collaborators: John Yang

This Assignment

One of the purposes of this homework is to help you diagnose your preparedness for the course. The rest of this course will assume familiarity with the programming and math concepts covered in this homework. If you struggle with this homework, please consider reviewing prerequisite material.

Score Breakdown

Question	Points
1	1
2a	1
2b	1
2c	1
2d	1
3	4
4a	3
4b	2
4c	2
4d	2

Question	Points
4e	2
5	2
6	4
7	2
8	4
Total	32

Here are some useful Jupyter notebook keyboard shortcuts. To learn more keyboard shortcuts, go to **Help -> Keyboard Shortcuts** in the menu above.

Here are a few we like:

1. `ctrl` + `return` : *Evaluate the current cell*
2. `shift` + `return` : *Evaluate the current cell and move to the next*
3. `esc` : *command mode* (may need to press before using any of the commands below)
4. `a` : *create a cell above*
5. `b` : *create a cell below*
6. `dd` : *delete a cell*
7. `m` : *convert a cell to markdown*
8. `y` : *convert a cell to code*

Initialize your environment

This cell should run without error if you're using the course Jupyter Hub or you have [set up your personal computer correctly (http://www.ds100.org/sp19/setup)](http://www.ds100.org/sp19/setup).

In [63]:
```
import numpy as np
import matplotlib
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
```

Python

Question 1

Recall the formula for population variance below:

$$\sigma^2 = \frac{\sum_{i=1}^{N}(x_i - \mu)^2}{N}$$

Complete the functions below to compute the population variance of `population` , an array of numbers. For this question, do not use built in NumPy functions; we will use NumPy to verify your code.

In [64]:
```
def mean(population):
    """
    Returns the mean of population (mu)

    Keyword arguments:
    population -- a numpy array of numbers
    """
    # Calculate the mean of a population
    sum = 0
    for elem in population:
        sum = sum + elem
    return sum / len(population)

def variance(population):
    """
    Returns the variance of population (sigma squared)

    Keyword arguments:
    population -- a numpy array of numbers
    """
    # Calculate the variance of a population
    sum = 0
    avg = mean(population)
    for elem in population:
        sum = sum + (elem - avg)**2
    return sum / len(population)
```

In [65]:
```
ok.grade("q1");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 2
    Failed: 0
[ooooooooook] 100.0% passed
```

NumPy

You should be able to understand the code in the following cells. If not, review the following:

- [DS100 NumPy Review (http://ds100.org/fa17/assets/notebooks/numpy/Numpy_Review.html)](http://ds100.org/fa17/assets/notebooks/numpy/Numpy_Review.html)
- [Condensed NumPy Review (http://cs231n.github.io/python-numpy-tutorial/#numpy)](http://cs231n.github.io/python-numpy-tutorial/#numpy)
- [The Official NumPy Tutorial (https://docs.scipy.org/doc/numpy-dev/user/quickstart.html)](https://docs.scipy.org/doc/numpy-dev/user/quickstart.html)

- [The Data 8 Textbook Chapter on NumPy (https://www.inferentialthinking.com/chapters/05/1/Arrays)](https://www.inferentialthinking.com/chapters/05/1/Arrays)

Jupyter pro-tip: Pull up the docs for any function in Jupyter by running a cell with the function name and a `?` at the end:

In [66]:
```
np.arange?
```

You can close the window at the bottom by pressing `esc` several times.

Another Jupyter pro-tip: Pull up the docs for any function in Jupyter by typing the function name, then `<Shift>-<Tab>` on your keyboard. This is super convenient when you forget the order of the arguments to a function. You can press `<Tab>` multiple times to expand the docs and reveal additional information.

Try it on the function below:

In [67]:
```
np.linspace
```

Out[67]:
```
<function numpy.core.function_base.linspace(start, stop, num=50, endpoint
=True, retstep=False, dtype=None)>
```

Now, let's go through some linear algebra coding questions with NumPy. In this question, we'll ask you to use your linear algebra knowledge to fill in NumPy matrices. To conduct matrix multiplication in NumPy, you should write code like the following:

In [68]:
```
# A matrix in NumPy is a 2-dimensional NumPy array
matA = np.array([
    [1, 2, 3],
    [4, 5, 6],
])

matB = np.array([
    [10, 11],
    [12, 13],
    [14, 15],
])

# The notation B @ v means: compute the matrix multiplication Bv
matA @ matB
```

Out[68]:
```
array([[ 76,  82],
       [184, 199]])
```

You can also use the same syntax to do matrix-vector multiplication or vector dot products. Handy!

In [69]:
```
matA = np.array([
    [1, 2, 3],
    [4, 5, 6],
])

# A vector in NumPy is simply a 1-dimensional NumPy array
some_vec = np.array([ 10, 12, 14, ])

another_vec = np.array([ 10, 20, 30 ])

print(matA @ some_vec)
print(some_vec @ another_vec)
```

```
[ 76 184]
760
```

Question 2a

Joey, Deb, and Sam are shopping for fruit at Berkeley Bowl. Berkeley Bowl, true to its name, only sells fruit bowls. A fruit bowl contains some fruit and the price of a fruit bowl is the total price of all of its individual fruit.

Berkeley Bowl has apples for \$2.00, bananas for \$1.00, and cantaloupes for \$4.00 (expensive!). The price of each of these can be written in a vector:

$$\vec{v} = \begin{bmatrix} 2 \\ 1 \\ 4 \end{bmatrix}$$

Berkeley Bowl sells the following fruit bowls:

1. 2 of each fruit
2. 5 apples and 8 bananas
3. 2 bananas and 3 cantaloupes
4. 10 cantaloupes

Create a 2-dimensional numpy array encoding the matrix B such that the matrix-vector multiplication

$$B\vec{v}$$

evaluates to a length 4 column vector containing the price of each fruit bowl. The first entry of the result should be the cost of fruit bowl #1, the second entry the cost of fruit bowl #2, etc.

In [70]:
```
v = np.array([2,1,4])

B = np.array([
    [2, 2, 2],
    [5, 8, 0],
    [0, 2, 3],
    [0, 0, 10]
])

# The notation B @ v means: compute the matrix multiplication Bv
B @ v
```

Out[70]:
```
array([14, 18, 14, 40])
```

In [71]:
```
ok.grade("q2a");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 2
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 2b

Joey, Deb, and Sam make the following purchases:

- Joey buys 2 fruit bowl #1s and 1 fruit bowl #2.
- Deb buys 1 of each fruit bowl.
- Sam buys 10 fruit bowl #4s (he really like cantaloupes).

Create a matrix A such that the matrix expression

$$AB\vec{v}$$

evaluates to a length 3 column vector containing how much each of them spent. The first entry of the result should be the total amount spent by Joey, the second entry the amount sent by Deb, etc.

Note that the tests for this question do not tell you whether your answer is correct. That's up to you to determine.

In [72]:
```
A = np.array([
    [2, 1, 0, 0],
    [1, 1, 1, 1],
    [0, 0, 0, 10]
])

A @ B @ v
```

Out[72]:
```
array([ 46,  86, 400])
```

In [73]:
```
ok.grade("q2b");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 2c

Who spent the most money? Assign `most` to a string containing the name of this person.

In [74]:
```
most = "Sam"
```

In [75]:
```
ok.grade("q2c");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Question 2d

Let's suppose Berkeley Bowl changes their fruit prices, but you don't know what they changed their prices to. Joey, Deb, and Sam buy the same quantity of fruit baskets and the number of fruit in each basket is the same, but now they each spent these amounts:

$$\vec{x} = \begin{bmatrix} 80 \\ 80 \\ 100 \end{bmatrix}$$

Use `np.linalg.inv` and the above final costs to compute the new prices for the individual fruits as a vector called `new_v`.

In [76]:
```
x = np.array([80,80,100])
new_v = np.linalg.inv(A @ B) @ x
new_v
```

Out[76]:
```
array([5.5       , 2.20833333, 1.        ])
```

In [77]:
```
ok.grade("q2d");
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Running tests

---------------------------------------------------------------------
Test summary
    Passed: 1
    Failed: 0
[ooooooooook] 100.0% passed
```

Multivariable Calculus, Linear Algebra, and Probability

The following questions ask you to recall your knowledge of multivariable calculus, linear algebra, and probability. We will use some of the most fundamental concepts from each discipline in this class, so the following problems should at least seem familiar to you.

For the following problems, you should use LaTeX to format your answer. If you aren't familiar with LaTeX, not to worry. It's not hard to use in a Jupyter notebook. Just place your math in between dollar signs:

`$ f(x) = 2x $` becomes $f(x) = 2x$.

If you have a longer equation, use double dollar signs to place it on a line by itself:

`$$ \sum_{i=0}^n i^2 $$` becomes:

$$\sum_{i=0}^n i^2$$

.

Here is some handy notation:

Output	Latex
x^{a+b}	`x^{a + b}`
x_{a+b}	`x_{a + b}`
$\frac{a}{b}$	`\frac{a}{b}`
$\sqrt{a + b}$	`\sqrt{a + b}`

Output	Latex
$\{\alpha, \beta, \gamma, \pi, \mu, \sigma^2\}$	`\{ \alpha, \beta, \gamma, \pi, \mu, \sigma^2 \}`
$\sum_{x=1}^{100}$	`\sum_{x=1}^{100}`
$\frac{\partial}{\partial x}$	`\frac{\partial}{\partial x}`
$\begin{bmatrix} 2x + 4y \\ 4x + 6y^2 \\ \end{bmatrix}$	`\begin{bmatrix} 2x + 4y \\ 4x + 6y^2 \\ \end{bmatrix}`

For more about basic LaTeX formatting, you can read this article. (https://www.sharelatex.com/learn/Mathematical_expressions)

If you have trouble with these topics, we suggest reviewing:

- Khan Academy's Multivariable Calculus (https://www.khanacademy.org/math/multivariable-calculus)
- Khan Academy's Linear Algebra (https://www.khanacademy.org/math/linear-algebra)
- Khan Academy's Statistics and Probability (https://www.khanacademy.org/math/statistics-probability)

Recall that summation (or sigma notation) is a way of expressing a long sum in a concise way. Let $a_1, a_2, \ldots, a_n \in \mathbb{R}$ and $x_1, x_2, \ldots, x_n \in \mathbb{R}$ be collections of real numbers. When you see x_i, you can think of the i as an index for the i^{th} x. For example x_2 is the second x value in the list $x_1, x_2, \ldots, x_n$. We define sigma notation as follows:

$$\sum_{i=1}^{n} a_i x_i = a_1 x_1 + a_2 x_2 + \ldots + a_n x_n$$

We commonly use sigma notation to compactly write the definition of the arithmetic mean (commonly known as the `average`):

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + \ldots + x_n) = \frac{1}{n} \sum_{i=1}^{n} x_i$$

Question 3

For each of the statements below, either prove that it is true by using definitions or show that it is false by providing a counterexample.

Statement I

$$\frac{\sum_{i=1}^n a_i x_i}{\sum_{i=1}^n a_i} = \sum_{i=1}^n x_i$$

This is false. If we let $n = 3$, and $a_1 = 1, a_2 = 2, a_3 = 3$, and $x_1 = -1, x_2 = -2, x_3 = -3$...

The LHS evaluates to...
$\frac{-1 + -4 + -9}{1 + 2 + 3} = \frac{-14}{6} = \frac{-7}{3}$
The RHS evaluates to...
-1 + -2 + -3 = -6
As we can see, the LHS $\neq$ RHS.

Statement II

$\sum_{i=1}^{n} x_1 = nx_1$

From above, we can see that...
$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$
$\sum_{i=1}^{n} x_1 = n * \bar{x}$
$\sum_{i=1}^{n} x_1 = n * \frac{1}{n}(x_1 + x_1 + \ldots + x_n)$
$\sum_{i=1}^{n} x_1 = nx_1$

Statement III

$\sum_{i=1}^{n} a_3 x_i = na_3 \bar{x}$

From above, we can see that...
$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$
$na_3 \bar{x} = n\frac{1}{n}a_3 \ (x_1 + x_2 + \ldots + x_n)$
$na_3 \bar{x} = a_3 \sum_{i=1}^{n} x_i$
$na_3 \bar{x} = \sum_{i=1}^{n} a_3 x_i$

Statement IV

$\sum_{i=1}^{n} a_i x_i = n\bar{a}\bar{x}$

This is false. If we let $n = 3$, and $a_1 = 1, a_2 = 2, a_3 = 3$, and $x_1 = -1, x_2 = -2, x_3 = -3$...
The LHS evaluates to...
$-1 + -4 + -9 = -14$
The RHS evaluates to...
$3 \frac{(1 + 2 + 3)}{3} \frac{(-1 + -2 + -3)}{3} = (3)(2)(-2) = -12$
As we can see, the LHS $\neq$ RHS.

Note:

We can also generalize the summation concepts above to multiple indices: consider an array of values x_{ij}

$$\begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,n} \\ x_{2,1} & x_{2,2} & \dots & x_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,n} \end{bmatrix}$$

By convention, the first index refers to the row and the second index references the column. e.g. $x_{2,4}$ is the value in the second row and the fourth column. For multi-indexed arrays like this, we can write down the sum of all the values by evoking sigma notation multiple times:

$$\begin{aligned}
\sum_{i=1}^{n}\sum_{j=1}^{n} x_{i,j} &= \sum_{i=1}^{n}\left(\sum_{j=1}^{n} x_{i,j}\right) \\
&= \sum_{i=1}^{n}\left(x_{i,1} + x_{i,2} + \dots + x_{i,n}\right) \\
&= \sum_{i=1}^{n} x_{i,1} + \sum_{i=1}^{n} x_{i,2} + \dots + \sum_{i=1}^{n} x_{i,n} \\
&= \left(x_{1,1} + x_{1,2} + \dots + x_{1,n}\right) + \left(x_{2,1} + x_{2,2} + \dots + x_{2,n}\right) + \dots + \left(x_{n,1} + x_{n,2} + \dots + x_{n,n}\right) \\
&= x_{1,1} + x_{1,2} + \dots + x_{1,n} + x_{2,1} + x_{2,2} + \dots + x_{2,n} + \dots + \dots + x_{n,1} + x_{n,2} + \dots + x_{n,n}
\end{aligned}$$

Question 4a

Suppose we have the following scalar-valued function on x and y:

$$f(x, y) = x^2 + 4xy + 2y^3 + e^{-3y} + \ln(2y)$$

Compute the partial derivative of $f(x, y)$ with respect to x.

$\frac{\partial}{\partial x} = 2x + 4y$

Now compute the partial derivative of $f(x, y)$ with respect to y:

$\frac{\partial}{\partial y} = 4x + 6y^2 - 3e^{-3y} + \dfrac{1}{x}$

Finally, using your answers to the above two parts, compute $\nabla f(x, y)$ (the gradient of $f(x, y)$) and evaluate the gradient at the point ($x = 2, y = -1$).

$\nabla f(x, y) = \frac{df}{dx}(x, y) + \frac{df}{dy}(x, y)$
$\nabla f(x, y) = 2(2) + 4(-1) + 4(2) + 6(-1)^2 - 3e^3 + 1/2$
$\nabla f(x, y) = 4 - 4 + 8 + 6 - 3e^3 + 1/2$
$\nabla f(x, y) = 14.5 - 3e^3$

Question 4b

Find the value(s) of x which minimizes the expression below. Justify why it is the minimum.

$\sum_{i=1}^{10}(i-x)^2$

Let $\sigma(x) = \sum_{i=1}^{10}(i-x)^2$.
$\frac{d}{dx}\sigma(x) = -2\sum_{i=1}^{10}(i-x) = 0$
$\sum_{i=1}^{10}(i-x) = 0$
$\sum_{i=1}^{10} i - \sum_{i=1}^{10} x = 0$
$\sum_{i=1}^{10} i = \sum_{i=1}^{10} x$
$55 = 10x$
$x = 5.5$

Question 4c

Let $\sigma(x) = \dfrac{1}{1+e^{-x}}$. Show that $\sigma(-x) = 1 - \sigma(x)$.

Left Hand Side (LHS)
$\sigma(-x) = \dfrac{1}{1+e^{x}}$
$\sigma(-x) = \dfrac{1}{1+e^{x}} * \dfrac{1-e^{-x}}{1-e^{-x}} = \dfrac{1-e^{-x}}{1-e^{-x}+e^{x}-e^{0}} = \dfrac{1-e^{-x}}{e^{x}-e^{-x}} * \dfrac{-1}{-1} = \dfrac{e^{-x}-1}{e^{-x}-e^{x}}$
Right Hand Side (RHS)
$1-\sigma(x) = 1 - \dfrac{1}{1+e^{-x}}$
$1-\sigma(x) = \dfrac{1+e^{-x}-1}{1+e^{-x}} = \dfrac{e^{-x}}{1+e^{-x}} * \dfrac{1-e^{x}}{1-e^{x}} = \dfrac{e^{-x}-e^{0}}{1-e^{x}+e^{-x}-1} = \dfrac{e^{-x}-1}{e^{-x}-e^{x}}$
Since the LHS = RHS, we can see that

$$\sigma(-x) = 1 - \sigma(x)$$

.

Question 4d

Show that the derivative can be written as:

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$$

Left Hand Side (LHS)
$\frac{d}{dx}\sigma(x) = \frac{d}{dx}(1+e^{-x})^{-1}$
$\frac{d}{dx}\sigma(x) = -(1+e^{-x})^{-2}(-e^{-x})$
$\frac{d}{dx}\sigma(x) = \dfrac{e^{-x}}{(1+e^{-x})^2}$

Right Hand Side (RHS)

$\sigma(x)(1-\sigma(x)) = \frac{1}{1+e^{-x}}(1-\frac{1}{1+e^{-x}})$

$= \frac{1}{1+e^{-x}} - \frac{1}{(1+e^{-x})^2}$

$= \frac{1+e^{-x}-1}{(1+e^{-x})^2}$

$= \frac{e^{-x}}{(1+e^{-x})^2}$

Therefore, since LHS = RHS, we see that

$$\frac{d}{dx}\sigma(x) = \sigma(x)(1-\sigma(x))$$

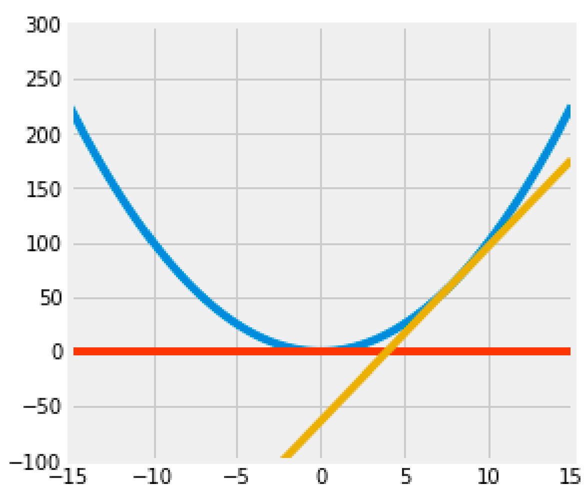
.

Question 4e

Write code to plot the function $f(x) = x^2$, the equation of the tangent line passing through $x = 8$, and the equation of the tangent line passing through $x = 0$.

Set the range of the x-axis to (-15, 15) and the range of the y axis to (-100, 300) and the figure size to (4,4).

Your resulting plot should look like this:



You should use the `plt.plot` function to plot lines. You may find the following functions useful:

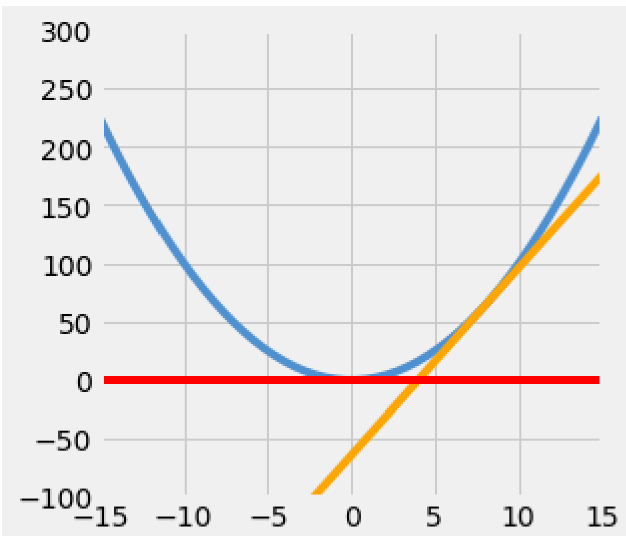
- `plt.plot(..)` (https://matplotlib.org/api/_as_gen/matplotlib.pyplot.plot.html)
- `plt.figure(figsize=..)` (https://stackoverflow.com/questions/332289/how-do-you-change-the-size-of-figures-drawn-with-matplotlib)
- `plt.ylim(..)` (https://matplotlib.org/api/_as_gen/matplotlib.pyplot.ylim.html)
- `plt.axhline(..)` (https://matplotlib.org/api/_as_gen/matplotlib.pyplot.hlines.html)

In [82]:
```
def f(x):
    return x**2

def df(x):
    return 16*x - 64

def plot(f, df):
    plt.plot(xRange, f(xRange), color='#4f90d0')
    plt.plot(xRange, df(xRange), color='orange')
    plt.plot(xRange, xRange*0, color='red')

xRange = np.linspace(-15, 15)
plt.figure(figsize=(4,4))
plt.axis([0, 15, -100, 300])
plt.xticks(np.linspace(-15, 15, num=7))
plot(f, df)
```



Question 5

Consider the following scenario:

Only 1% of 40-year-old women who participate in a routine mammography test have breast cancer. 80% of women who have breast cancer will test positive, but 9.6% of women who don't have breast cancer will also get positive tests.

Suppose we know that a woman of this age tested positive in a routine screening. What is the probability that she actually has breast cancer?

Hint: Use Bayes' rule.

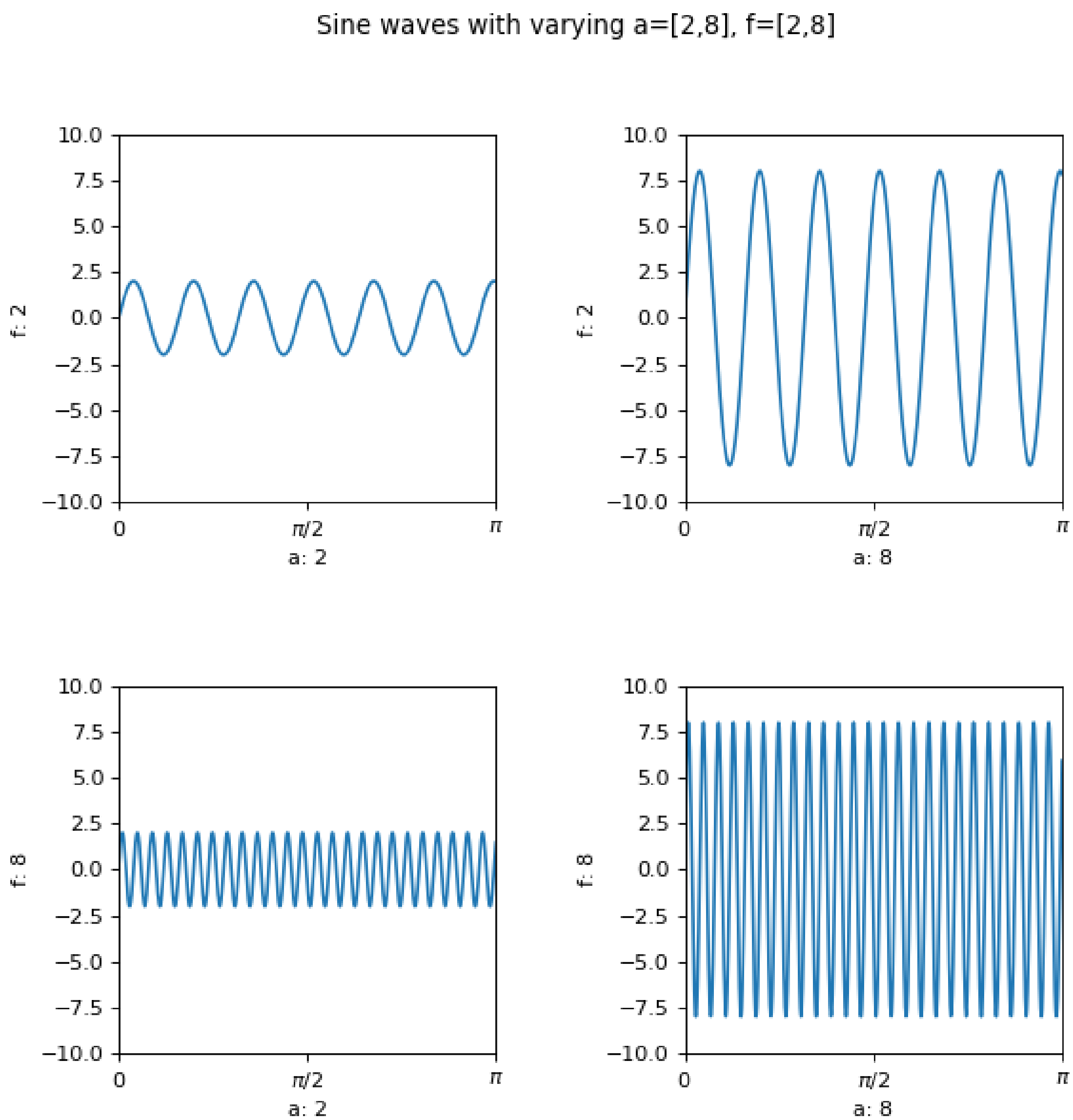
Pr(cancer_when_positive) = $\dfrac{\Pr(positive_when_cancer) * \Pr(cancer)}{\Pr(positive)}$

$$= \frac{0.01 * 0.8}{(0.01 * 0.8) + (0.99 * 0.096)}$$
$$= \frac{0.008}{0.10304}$$
$$= 7.76\%$$

Plotting

The following questions will allow you to gain more practice with the plotting library `matplotlib`.

Suppose we want to visualize the function $g(t) = a \cdot \sin(2\pi f t)$ while varying the values f, a.



Question 6

Generate a 2 by 2 plot that plots the function $g(t)$ as a line plot with values $f = 2, 8$ and $a = 2, 8$. Since there are 2 values of f and 2 values of a there are a total of 4 combinations, hence a 2 by 2 plot. The rows should vary in a and the columns should vary in f.

Set the x limit of all figures to $[0, \pi]$ and the y limit to $[-10, 10]$. The figure size should be 8 by 8. Make sure to label your x and y axes with the appropriate value of f or a. Additionally, make sure the x ticks are labeled $[0, \frac{\pi}{2}, \pi]$. Your overall plot should look something like the one above.

Hint 1: Modularize your code and use loops.

Hint 2: Are your plots too close together such that the labels are overlapping with other plots? Look at the [`plt.subplots_adjust` (https://matplotlib.org/api/_as_gen/matplotlib.pyplot.subplots_adjust.html)](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.subplots_adjust.html) function.

Hint 3: Having trouble setting the x-axis ticks and ticklabels? Look at the [`plt.xticks` (https://matplotlib.org/api/_as_gen/matplotlib.pyplot.xticks.html)](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.xticks.html) function.

Hint 4: You can add title to overall plot with `plt.suptitle` .

In [83]:
```
def g(t, a, f):
    return a * np.sin(2 * np.pi * f * t)

xRange = np.linspace(0, np.pi, num=500)
plt.figure(figsize=(8,8))
plt.suptitle("Sine waves with varying a=[2,8], f=[2,8]")
combinations = [(2, 2), (8, 2), (2, 8), (8, 8)]
i = 0
subInd = 1

for tuple in combinations:
    subplot = g(xRange, combinations[i][0], combinations[i][1])
    plt.xlabel("a: " + str(combinations[i][0]))
    plt.ylabel("f: " + str(combinations[i][1]))
    plt.axis([0, np.pi, -10, 10])
    plt.subplot(2, 2, subInd, facecolor='white')
    plt.subplots_adjust(wspace=0.5, hspace=0.5)
    plt.grid(b=None)
    plt.plot(xRange, subplot)
    plt.xticks([0, np.pi/2, np.pi], ['0', 'π/2', 'π'])
    plt.yticks(np.linspace(-10, 10, 9))
    subInd = subInd + 1
    i = i + 1
```

Question 7

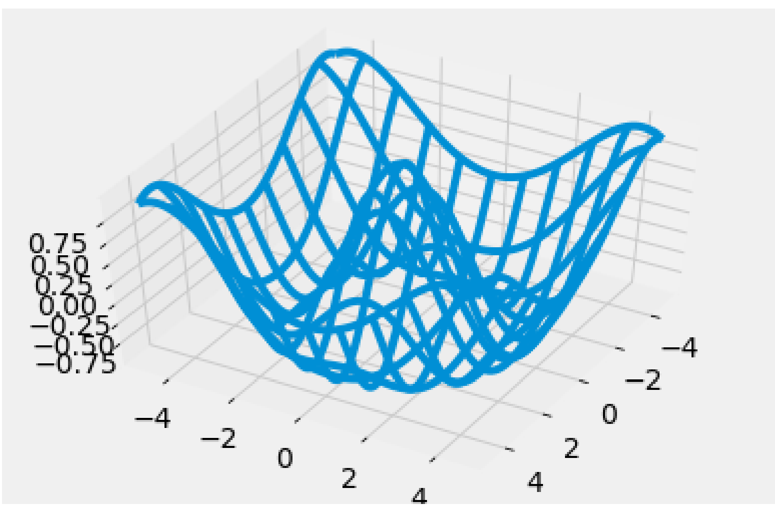
We should also familiarize ourselves with looking up documentation and learning how to read it. Below is a section of code that plots a basic wireframe. Replace each `# Your answer here` with a description of what the line above does, what the arguments being passed in are, and how the arguments are used in the function. For example,

```
np.arange(2, 5, 0.2)
# This returns an array of numbers from 2 to 5 with an interval siz
e of 0.2
```

Hint: The `Shift + Tab` tip from earlier in the notebook may help here. Remember that objects must be defined in order for the documentation shortcut to work; for example, all of the documentation will show for method calls from `np` since we've already executed `import numpy as np`. However, since `z` is not yet defined in the kernel, `z.reshape()` will not show documentation until you run the line `z = np.cos(squared)`.

In [84]:
```
from mpl_toolkits.mplot3d import axes3d

u = np.linspace(1.5*np.pi, -1.5*np.pi, 100)
# This sets the value of u to be 100 evenly spaced points, calculated over
[x,y] = np.meshgrid(u, u)
# Sets [x,y] to be coordinate matrices from coordinate vectors (u, u).
squared = np.sqrt(x.flatten()**2 + y.flatten()**2)
z = np.cos(squared)
# Takes the cosine of each element in @param: squared.
z = z.reshape(x.shape)
# Returns an array containing the same data with a new x.shape.
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
# Adds a 3-dimensional axis to the figure as part of a subplot arrangement,
# where pos is a three digit integer, such that the first digit (1) is the
# the second digit (1) is the number of columns, and the third digit (1) is
ax.plot_wireframe(x, y, z, rstride=10, cstride=10)
# Plots a 3-D wireframe using the x, y, z axes. The rstride (array row step
# arguments will determine at most how many evenly spaced samples (in this
ax.view_init(elev=50., azim=30)
# Set the elevation to 50 and azimuth to 30 of the axes.
plt.savefig("figure1.png")
# Save the current figure, in this case figure1.png.
```



Question 8

For a data-driven question of your choice, describe your approach and thought process in addressing this question. Outline what a sensible workflow might look like, including framing the question and identifying relevant data. Also consider transversal issues such as ethics and governance with respect to your question.

This question is about data-driven reasoning; you should focus more on *what* to do, than on *how* do to it exactly. You may use any of the questions presented in the first lecture, excluding the real estate, crowd size, and COMPAS questions. A complete response should contain about 250 words.

The data-driven question I would like to discuss is the Coordinated Entry System (CES) in Los Angeles, which attempts to find housing for the homeless. The problem with this system is that it collects fairly personal information about these homeless people (such as immigration and residency status, domestic violence history, health and mental health histories, substance use, sexuality, etc.). Afterward, a digital registry stores this data and runs an algorithm to determine a score which, in turn, is used to determine whether a homeless person can receive housing. The problem with this system is that even after divulging such personal information, there is no guarantee that a homeless person will receive housing. Furthermore, there is no guarantee that the CES will wipe away all the data stored on a particular individual. So, if a homeless person doesn't receive housing, he or she would have divulged most, if not all, of his or her personally identifiable information for absolutely nothing in return. I believe that a sensible workflow should hold the registry accountable for withholding homeless people's data. A reasonable question to ask is "Of all the homeless people who submit their personal information, how many actually receive housing offers and for the proportion that don't, are they concerned that they revealed all their information for nothing in return?" A good starting point would be to conduct a survey on the people who did and didn't receive housing and see whether this result was positively or negatively correlated with whether or not they were bothered by providing their personal information.

In [85]:
```
# Save your notebook first, then run this cell to submit.
import jassign.to_pdf
jassign.to_pdf.generate_pdf('hw1.ipynb', 'hw1.pdf')
ok.submit()
```

```
Generating PDF...
Saved hw1.pdf

Saving notebook... Saved 'hw1.ipynb'.
Submit... 100% complete
Submission successful for user: kushal.singh@berkeley.edu
URL: https://okpy.org/cal/data100/sp19/hw1/submissions/xvZG2n (https://ok
py.org/cal/data100/sp19/hw1/submissions/xvZG2n)
```

In [ ]: