Notebook

September 2, 2019

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

Let a= [1,2,3] x = [3,6,9] Hence LHS = $\frac{\sum_{i=1}^{n} a_i x_i}{\sum_{i=1}^{n} a_i} = \frac{1*3+2*6+3*9}{1+2+3} = 7$

RHS = $\sum_{i=1}^{n} x_i = 3+6+9 = 18 \neq 7$

Hence LHS $\neq$ RHS, false

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

True $\sum_{i=1}^{n} x_1 = x_1 + x_1 + x_1 + ...$ (n times) $= n * x_1$

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

RHS $= n a_3 \bar{x} = n a_3 \frac{1}{n} \sum_{i=1}^{n} x_i = a_3 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} a_3 x_i =$ LHS.. Hence True

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

RHS $= n\bar{a}\bar{x} = n\dfrac{1}{n}\sum_{i=1}^{n} a_i \dfrac{1}{n}\sum_{i=1}^{n} x_i = \dfrac{1}{n}\sum_{j=1}^{n}\sum_{i=1}^{n} a_j x_i \neq$ LHS

as when $a = [1, 2, 3]$ and $x = [4, 5, 6]$ the $\frac{1}{n}\sum_{i=1}^{n}\sum_{i=1}^{n} a_i x_i$ is 30 where as $\sum_{i=1}^{n} a_i x_i$ is 32. Hence False

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$x$.
$\frac{\partial f(x,y)}{\partial x} = 2x + 4y$

Now compute the partial derivative of $f(x, y)$ with respect to y:
$\frac{\partial f(x,y)}{\partial y} = 4x + 6y^2 - 3e^{-3y} + \dfrac{1}{y}$

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 $(x=2, y=-1)$.

$\nabla f(x,y) =< \frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y} >=< 0, -3e^{-3y} + 13 >$

Question 4b Find the value(s) of x which minimizes the expression below. Justify why it is the minimum.

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

$\frac{d\sum_{i=1}^{10}(i-x)^2}{dx} = \sum_{i=1}^{10} -2(i-x) = 0 \implies \sum_{i=1}^{10} -2i + 2x = -110 + 20x = 0 \implies x = 110/20 = 5.5$

Question 4c Let $\sigma(x) = \dfrac{1}{1+e^{-x}}$. Show that $\sigma(-x) = 1 - \sigma(x)$.

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

$RHS = 1 - \sigma(x) = 1 - \dfrac{1}{1+e^{-x}} = 1 - \dfrac{e^{x}}{1+e^{x}} = \dfrac{1+e^{x}-e^{x}}{1+e^{x}} = \dfrac{1}{1+e^{x}} = LHS$

Hence Proved

Question 4d Show that the derivative can be written as:

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

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

$RHS = \sigma(x)(1-\sigma(x)) = \sigma(x)\sigma(-x)$ (from above) $= \dfrac{1}{1+e^{-x}}\dfrac{1}{1+e^{x}} = \dfrac{e^{x}}{(1+e^{x})^2} = RHS$

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:

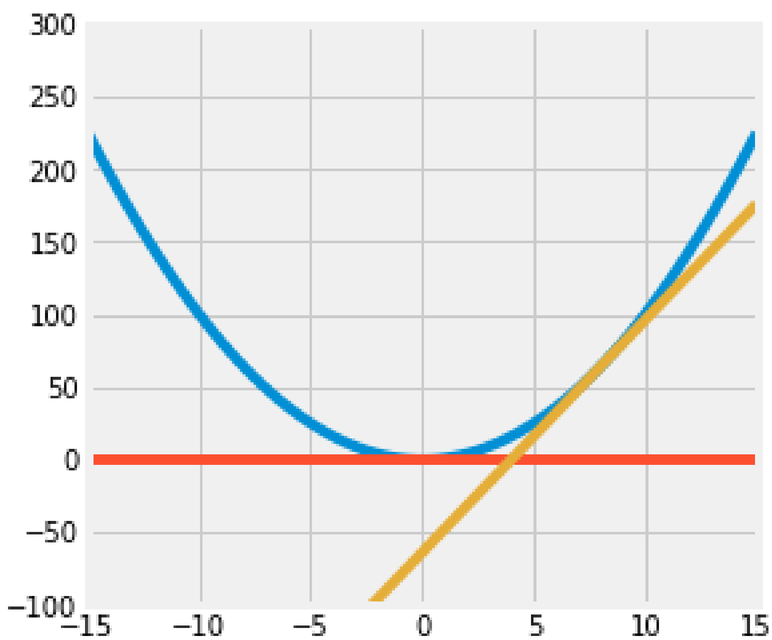
- `plt.plot(..)`
- `plt.figure(figsize=..)`
- `plt.ylim(..)`
- `plt.axhline(..)`

In [17]:
```
def f(x):
    return x**2

def df(x):
    return 2*x

def plot(f, df):
    x = np.linspace(-15,15,200)
    y = f(x)
    y_0_1 = f(8)
    y_tan = df(8) * (x - 8) + y_0_1
    y_0_2 = f(0)
    y_tan_2 = df(0) * (x - 0) + y_0_2
    plt.figure(figsize=(4,4))
    plt.plot(x,y,'-')
    plt.plot(x,y_tan_2,'-')
    plt.plot(x,y_tan, '-')
    plt.ylim((-100,300))
    plt.xlim((-15,15))


plot(f, df)
```



0.0.1 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?

You **must** show work using LaTex (not code) to get credit for your answer.

Hint: Use Bayes' rule.

Let A be the event that the women have breast cancer. Let B be the probability that the test results are positive.

Hence P(A) = 0.01

P(B|A) = 0.8

P(B|-A) = 0.096

$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$

P(B) = P(A)* P(B|A)+P(-A)* P(B|-A) using law of Total Probability

Hence P(B) = 0.01*0.8+ 0.99*0.096 = 0.103

Hence P(A|B) = P(women tested positive has breast cancer) = $\frac{0.8 * 0.01}{0.103} = 0.078$

0.0.2 Question 6

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 size 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 [18]: from mpl_toolkits.mplot3d import axes3d

         u = np.linspace(1.5*np.pi, -1.5*np.pi, 100)
         #this returns a evenly spaced 100 points between 1.5pi and -1.5pi. It returns an array of numb
         [x,y] = np.meshgrid(u, u)
         #This returns a array of possible coordinate points using the vector u. From the documentation
         squared = np.sqrt(x.flatten()**2 + y.flatten()**2)
         z = np.cos(squared)
         # the flatten function returns a copy of the array in a one dimensional form. Then squared con
         z = z.reshape(x.shape)
         # This reshapes z to have the same shape as the x array

         fig = plt.figure()
         ax = fig.add_subplot(111, projection='3d')
         # this adds the subplot in the 1,1,1 position in the 3d axis. Here there is only one subplot a
         ax.plot_wireframe(x, y, z, rstride=10, cstride=10)
         # This adds the wireframe plot to the subplot established above and plots the z array against
         ax.view_init(elev=50., azim=30)
         # this sets the elevation angle and the azimuth for viewing the 3d graph
         plt.savefig("figure1.png")
         # this saves the figure as a png file
```