Notebook

June 25, 2019

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

Counterexample (False):

$$left = \frac{\begin{bmatrix}1 & 2 & 3\end{bmatrix}\begin{bmatrix}4\\5\\6\end{bmatrix}}{\begin{bmatrix}1 & 2 & 3\end{bmatrix}\begin{bmatrix}1\\1\\1\end{bmatrix}} = \frac{16}{3} \neq \begin{bmatrix}1 & 1 & 1\end{bmatrix}\begin{bmatrix}4\\5\\6\end{bmatrix} = 15 = right$$

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

Prove (True):

$$left = \sum_{i=1}^{n} x_1 = \underbrace{x_1 + x_1 + ... + x_1}_{\text{n}} = nx_1 = right$$

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

Prove (True):

$$left = \sum_{i=1}^{n} a_3 x_i = a_3 x_1 + a_3 x_2 + ... + a_3 x_n = a_3(x_1 + x_2 + ... + x_n) right = n a_3 \bar{x} = n a_3 \frac{1}{n}(x_1 + x_2 + ... + x_n) = a_3(x_1 + x_2 + ... + x$$

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

$$left = \sum_{i=1}^{n} a_i x_i = (a_1 x_1 + a_2 x_2 + ... + a_n x_n) right = n\bar{a}\bar{x} = n\frac{1}{n}\sum_{i=1}^{n} a_i \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}\begin{bmatrix} a_1 & a_2 & ... & a_n \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & ... & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = $$

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$x$.

$$f_x(x, y) = 2x + 4y since \frac{d}{dx}x^2 = 2x, \frac{d}{dx}x = 1$$

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

$$f_y(x, y) = 0 + 4x + 6y^2 - 3e^{-3y} + \frac{1}{y} = 4x + 6y^2 - 3e^{-3y} + \frac{1}{y} since \frac{d}{dy} y = 1, \frac{d}{dy} e^{-3y} = -3e^{-3y}, \frac{d}{dy} ln(2y) = \frac{1}{y}$$

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

$\nabla f(x,y) = \frac{\partial f}{\partial x} i + \frac{\partial f}{\partial y} j = (2x + 4y)i + (4x + 6y^2 - 3e^{-3y} + \frac{1}{y})j$

$Therefore, x = 2, y = -1, \nabla f(x,y) = (13 - 3e^3)j$

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

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

$\sum_{i=1}^{10}(i-x)^2 = 1-x)^2 + (2-x)^2 + ... + (10-x)^2 = 10x^2 - 110x + 385$. From this we can see $a = 10 > 0, b = -110, c = 385$, therefore when $x = \frac{-b}{2a} = 5.5$, the expression above is the smallest.

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

Because left = $\sigma(-x) = \frac{1}{1+e^{x}}$,right = $1 - \sigma(x) = 1 - \dfrac{1}{1+e^{-x}} = \frac{e^{-x}}{1+e^{-x}} = \frac{1}{\frac{1}{e^{-x}}+1} = \frac{1}{e^{x}+1}$,therefore, left=$\sigma(-x) = 1 - \sigma(x)$=right

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

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

Write your answer here, replacing this text.

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:

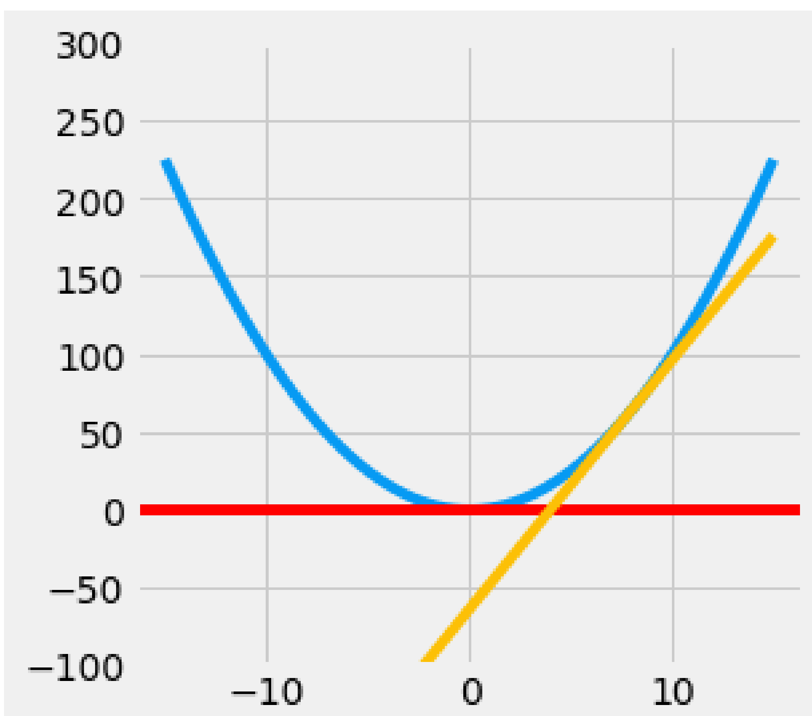
- `plt.plot(..)`
- `plt.figure(figsize=..)`
- `plt.ylim(..)`
- `plt.axhline(..)`

```
In [177]: def f(x):

              return x**2

          def df(x):
              h = 0.000000001
              return (f(x+h) - f(x))/h

          def plot(f, df):
              x = np.linspace(-15,15,200)
              x_8 = 8
              y_8 = f(x_8)
              y_tan = df(x_8) * (x - x_8) + y_8
              plt.figure(figsize=(4,4))
              plt.ylim((-100,300))
              plt.plot(x,f(x),'xkcd:azure')
              plt.axhline(0,-15,15,color = 'r')
              plt.plot(x,y_tan,'xkcd:marigold')
          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?

Hint: Use Bayes' rule.

$A : havingbreastcancer$
$B : testpositive$
$P(B|A) = 0.8$
$P(A) = 0.01$
$P(B) = 0.01 * 0.8 + 0.99 * 0.096 = 0.10304$
$therefore, P(A|B) = \frac{P(B|A)P(A)}{P(B)} = \frac{0.8*0.01}{0.10304} = \frac{0.008}{0.10304} \approx 0.078$

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,

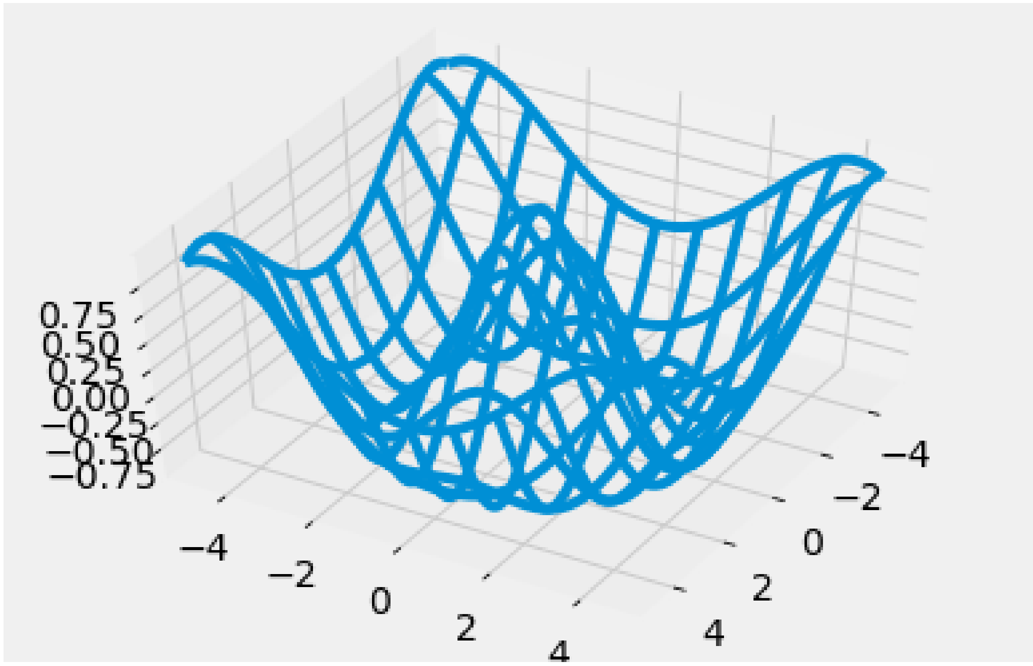
```
np.arange(2, 5, 0.2)
# This returns an array of numbers from 2 to 5 with an interval size of 0.2
```

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 [178]: from mpl_toolkits.mplot3d import axes3d

          u = np.linspace(1.5*np.pi, -1.5*np.pi, 100)
          # This returns an array of 100 evenly spaced samples, calculated over the interval [1.5*np.pi
          [x,y] = np.meshgrid(u, u)
          # Make 2-D coordinate arrays[x,y]for vectorized evaluations of 2-D scalar/vector fields over
          squared = np.sqrt(x.flatten()**2 + y.flatten()**2)
          z = np.cos(squared)
          # This returns an array of numbers by cosining squared, an array in radians with all the 1000
          z = z.reshape(x.shape)
          # This change the shape of the array z with the same data as before from (10000,) to the shap

          fig = plt.figure()
          ax = fig.add_subplot(111, projection='3d')
          # This creates an Axes3D object by adding a new axe to the figure, with 1 row, 1 column, and
          ax.plot_wireframe(x, y, z, rstride=10, cstride=10)
          # This plots a 3D wireframe using the data values of 2-D arrays X,Y, and Z, with the array ro
          ax.view_init(elev=50., azim=30)
          # Set the elevation angle in the z plane to 50.0 and the azimuth angle in the x,y plane to 30
          plt.savefig("figure1.png")
          # Save the current figure with the name "figure1.png".
```



0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−4
−2
0
2
4
−4
−2
0
2
4