Notebook

January 26, 2020

0.0.1 Question 1

Let $x_1, x_2, \ldots, x_n$ be a list of numbers. You can think of each index i as the label of a household, and the entry x_i as the annual income of Household i. Define the *mean* or *average* of the list to be $\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Question 1a) The ith *deviation from average* is the difference $x_i - \mu$. In Data 8 you saw in numerical examples that the sum of all these deviations is 0. Now prove that fact. That is, show that $\sum_{i=1}^{n}(x_i - \mu) = 0$.

Note: In this class, you must always put your answer in the cell that immediately follows the question. DO NOT create any cells between this one and the one that says *Write your answer here, replacing this text.*

$$
\begin{aligned}
&\sum_{i=1}^{n}(x_i - \mu) \\
=&\sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \mu \\
=&\sum_{i=1}^{n} x_i - n\mu \\
=&\sum_{i=1}^{n} x_i - n \times \frac{1}{n}\sum_{i=1}^{n} x_i \\
=&0
\end{aligned}
$$

Question 1b) Recall that the *variance* of a list is defined as the *mean squared deviation from average*, and that the *standard deviation* (SD) of the list is the square root of the variance. The SD is in the same units as the data and measures the rough size of the deviations from average.

Denote the variance of the list by σ^2. Write a math expression for σ^2. We recommend building your expression by reading the definition of variance from right to left. That is, start by writing the notation for "average", then "deviation from average", and so on.

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}[x_i - \frac{1}{n}\sum_{i=1}^{n}x_i]^2$$

Question 1c) Suppose you have to predict the value of x_i for some i, but you don't get to see i and you certainly don't get to see x_i. You decide that whatever x_i is, you're just going use you favorite number as your predictor, and your favorite number is μ.

The *error* in your prediction is $x_i - \mu$, which is your old friend the deviation from average. Thus the *mean squared error* (MSE) of your predictor μ over the entire list is the mean squared deviation from average, which is your old friend the variance. So we will write $\sigma^2 = MSE(\mu)$.

Now suppose I decide that whatever x_i is, I'm just going to use *my* favorite number as my predictor, and my favorite number is c. Write a math expression for $MSE(c)$. Again, go from right to left: first c, then the error, and so on.

$$MSE(c) = \frac{1}{n}\sum_{i=1}^{n}[x_i - c]^2$$

Question 1d) Whose predictor is better? It seems reasonable to guess that your predictor μ is better than my favorite but possibly weird c. Show that $MSE(c) > MSE(\mu)$ for all $c \neq \mu$, by the method indicated below.

- Write the error $x_i - c$ as $x_i - c = (x_i - \mu) + (\mu - c)$.
- Substitute this expression for $x_i - c$ in your formula for $MSE(c)$.
- Expand the square and use properties of sums; don't forget what you showed in Part **a**.

This shows that μ is the *least squares* constant predictor. In Data 8 you found (numerically) the least squares linear predictor of a variable y based on a related variable x. We will return to that later in this course, using a generalization of the calculation in this exercise.

$$
\begin{aligned}
&MSE(c) \\
=&\frac{1}{n}\sum_{i=1}^{n}[x_i - c]^2 \\
=&\frac{1}{n}\sum_{i=1}^{n}[(x_i - \mu) + (\mu - c)]^2 \\
=&\frac{1}{n}\sum_{i=1}^{n}[(x_i - \mu)^2 + (\mu - c)^2 + 2(x_i - \mu)(\mu - c)] \\
=&\frac{1}{n}\sum_{i=1}^{n}[x_i - \mu]^2 + \frac{1}{n}\sum_{i=1}^{n}[\mu - c]^2 + \frac{1}{n}\sum_{i=1}^{n}[2(x_i - \mu)(\mu - c)] \\
=&\frac{1}{n}\sum_{i=1}^{n}[x_i - \mu]^2 + \frac{1}{n}\sum_{i=1}^{n}[\mu - c]^2 + \frac{2(\mu - c)}{n}\sum_{i=1}^{n}[x_i - \mu] \\
=&\frac{1}{n}\sum_{i=1}^{n}[x_i - \mu]^2 + \frac{1}{n}\sum_{i=1}^{n}[\mu - c]^2 + \frac{2(\mu - c)}{n} \times 0 \\
=&\frac{1}{n}\sum_{i=1}^{n}[x_i - \mu]^2 + \frac{1}{n}\sum_{i=1}^{n}[\mu - c]^2 \\
>&\frac{1}{n}\sum_{i=1}^{n}[x_i - \mu]^2 \\
=&MSE(\mu)
\end{aligned}
$$

0.0.2 Question 2

Consider the function $f(x) = x^2$ for $-\infty < x < \infty$.

Question 2a) Find the equation of the tangent line to f at $x = 0$.

$f(x) = x^2$
$\therefore f(0) = 0, f'(x) = 2x$
$\therefore f'(0) = 0$
$\therefore$ tangent line is $y = 0$

Question 2b) Find the equation of the tangent line to f at $x = 8$.

$f(x) = x^2$

$\therefore f(8) = 64, f'(x) = 2x$

$\therefore f'(8) = 16$

$\therefore$ tangent line is $y = 16x - 64$

Question 2c) Write code to plot the function f, the tangent line at $x = 8$, and the tangent line at $x = 0$.

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).

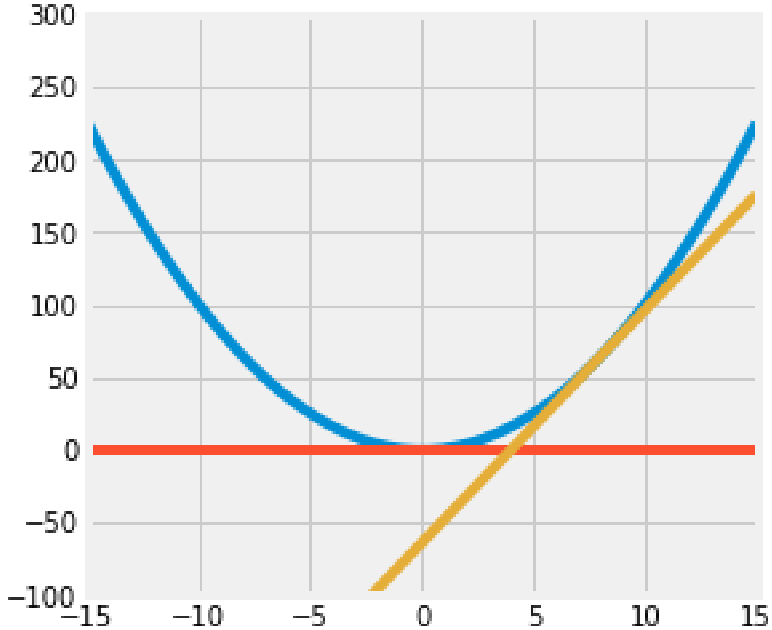
Your resulting plot should look like this:

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 [123]:
```
def f(x):
    return x**2

def df(x):
    return 16* -64

def plot(f, df):
    x=np.linspace(-15, 15)
    plt.figure(figsize=(4,4))
    plt.ylim(-100, 300)
    plt.xlim(-15, 15)

    # The fastest way
    # plt.plot(x,f(x),x,0*x,x,df(x))

    # Do it as problem requested
    # Get color cycle list
    # print(plt.rcParams['axes.prop_cycle'].by_key()['color'])
    plt.plot(x,f(x), c='#008fd5')
    plt.axhline(y=0, xmin=-15, xmax=15, c='#fc4f30')
    plt.plot(x,df(x), c='#e5ae38')

    return plt

plot(f, df)
```

Out[123]:
```
<module 'matplotlib.pyplot' from '/srv/conda/envs/data100/lib/python3.6/site-packages/matplot
```

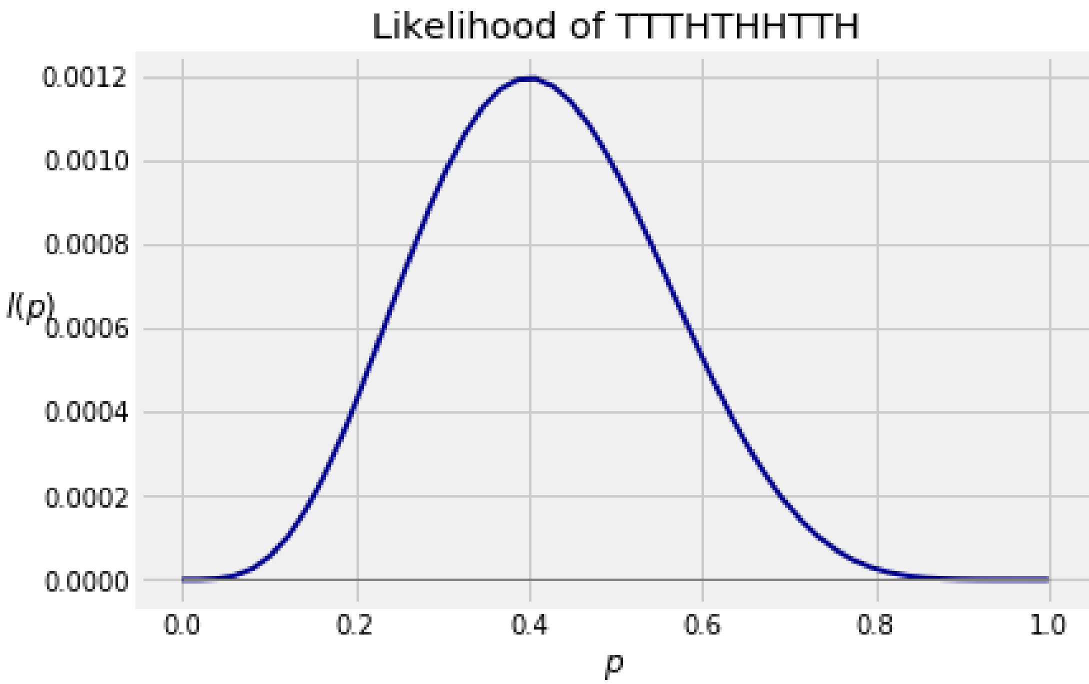

300
250
200
150
100
50
0
−50
−100
−15
−10
−5
0
5
10
15

Question 3b) I have a coin that lands heads with an unknown probability p. I toss it 10 times and get the sequence TTTHTHHTTH.

If you toss this coin 10 times, the chance that you get the sequence above is a function of p. That function is called the *likelihood* of the sequence TTTHTHHTTH, so we will call it l.

Plot the graph of l as a function of p for $p \in [0, 1]$.

```
In [3]: p = np.linspace(0, 1)
        likelihood = p**4*(1-p)**6
        plt.plot(p, likelihood, lw=2, color='darkblue') # lw is line width
        plt.plot([0, 1], [0, 0], lw=1, color='grey')     # horizontal axis
        plt.xlabel('$p$')
        plt.ylabel('$l(p)$', rotation=0)
        plt.title('Likelihood of TTTHTHHTTH');
```



Question 3c) The value $\hat{p}$ at which the likelihood function attains its maximum is called the *maximum likelihood estimate* (MLE) of p. Among all values of p, it is the one that makes the observed data most likely.

Please provide the value of $\hat{p}$ and also a simple interpretation of that value in terms of the data TTTHTHHTTH.

$\hat{p} = 0.4$

Interpretation: Among all values of p, it is the one $p = 0.6$ that makes the observed data TTTHTHHTTH most likely.

Question 3d) Explain why the value $\hat{p}$ at which the function l attains its maximum is the same as the value at which the function $\log(l)$ attains its maximum. To clarify, $\log(l)$ is the composition of log and l: $\log(l)$ at p is $\log(l(p))$. Even though it doesn't make a difference for this problem, log is now and forevermore the log to the base e, not to the base 10.

It might help to compare $\log(x_1)$ and $\log(x_2)$ for $x_1 < x_2$.

The observation in this exercise is hugely important in data science because many probabilities are products and the log function turns products into sums. It's much simpler to work with a sum than with a product.

$\because$ log(x) is strictly monotonically increasing

$\therefore \forall x_1 < x_2, log(x_1) < log(x_2); \forall log(x_1) < log(x_2), x_1 < x_2$

Given the value $\hat{p}$ at which the function l attains its maximum, assume a value $l(q) \neq l(p)$ at which the function $\log(l)$ attains its maximum

$\therefore log(l(q)) > log(l(p))$

$\therefore l(q) > l(p)$

$\because \hat{p}$ is the value at which the function l attains its maximum

$\therefore l(q) < l(p)$, which is contradictory

$\therefore$ the assumption is wrong, giving the conclusion that the value $\hat{p}$ at which the function l attains its maximum, is the same as the value at which the function $\log(l)$ attains its maximum.

Question 3e) Use Part c and calculus to find $\hat{p}$. Using Part d makes the calculus much easier. You don't have to check that the value you've found produces a max and not a min – we'll spare you that step.

using part c

$$\begin{aligned}
l &= p^6(1-p)^4 \\
l' &= 6p^5(1-p)^4 - 4p^6(1-p)^3 \\
&= p^5(1-p)^3(6-6p-4p) \\
&= p^5(1-p)^3(6-10p) \\
&= 0 \\
&\therefore p = 0.6
\end{aligned}$$

using part d

$$\begin{aligned}
l &= p^6(1-p)^4 \\
log(l) &= 6log(p) + 4log(1-p) \\
[log(l)]' &= \frac{6}{p} - \frac{4}{1-p} \\
&= 0 \\
&\therefore p = 0.6
\end{aligned}$$

0.0.3 Question 4

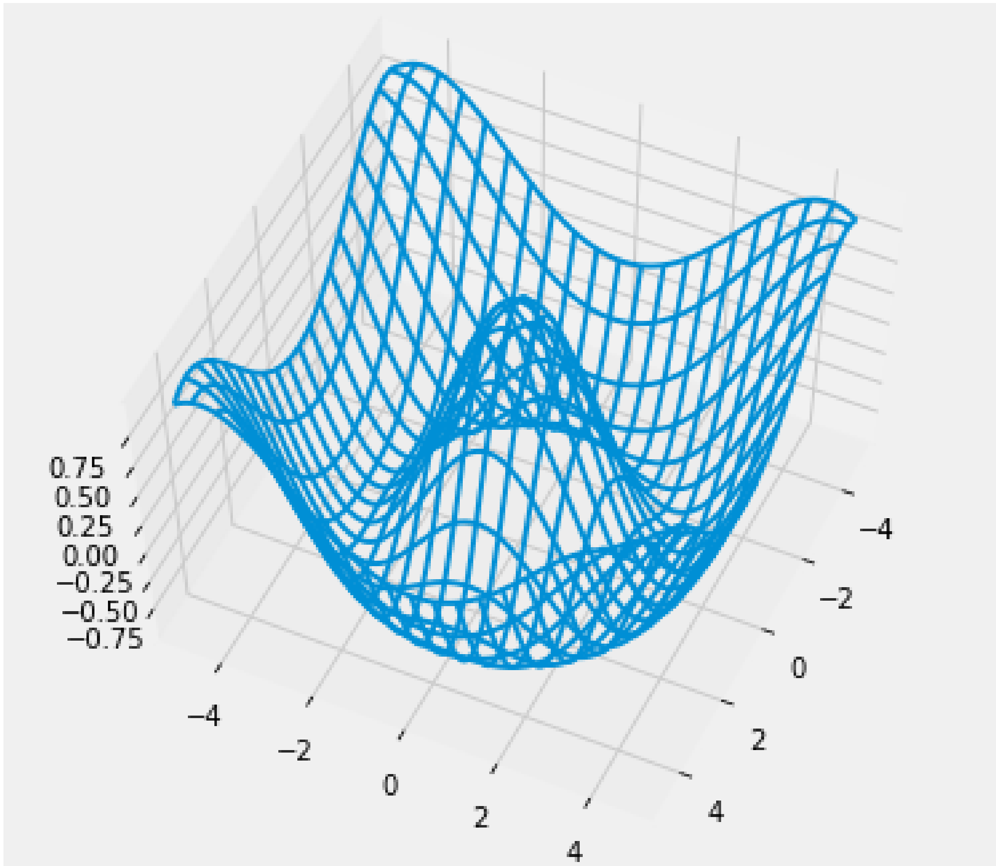
Data science is a rapidly expanding field and no degree program can hope to teach you everything that will be helpful to you as a data scientist. So it's important that you become familiar with looking up documentation and learning how to read it.

Below is a section of code that plots a three-dimensional ”wireframe” plot. You'll see what that means when you draw it. 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 [78]: from mpl_toolkits.mplot3d import axes3d

         u = np.linspace(1.5*np.pi, -1.5*np.pi, 100)
         # Define u as an array of 100 numbers from 1.5pi to -1.5 pi
         [x,y] = np.meshgrid(u, u)
         # Define [x,y] as coordinate matrices where x and y respectively recording the
         # x-coordinate and y-coordinate of mesh points, from coordinate vectors being both u.
         squared = np.sqrt(x.flatten()**2 + y.flatten()**2)
         z = np.cos(squared)
         # Define z = cos(sqrt((x^2+y^2))) for each point as an array flattened from coordinate
         # matrix
         z = z.reshape(x.shape)
         # Format the array z back to the shape of x (i.e, a matrix)

         fig = plt.figure(figsize=(6, 6))
         ax = fig.add_subplot(111, projection='3d')
         # Subplot the figure in a grid of row 1, col 1, at index 1, with 3d projection
         ax.plot_wireframe(x, y, z, rstride=5, cstride=5, lw=2)
         # Plot a 3D wirefram with row step 5, col step 5, line-width 2
         ax.view_init(elev=60, azim=25)
         # Set the elevation angle in the z plane elev=60 and the azimuth angle in the x,y
         # plane azim=25
         plt.savefig("figure1.png")
         # Save figure as "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

0.0.4 Question 5

Much of data analysis involves interpreting proportions – lots and lots of related proportions. So let's recall the basics. It might help to start by reviewing the main rules from Data 8, with particular attention to what's being multiplied in the multiplication rule.

Question 5a) The Pew Research Foundation publishes the results of numerous surveys, one of which is about the trust that Americans have in groups such as the military, scientists, and elected officials to act in the public interest. A table in the article summarizes the results.

Pick one of the options (i) and (ii) to answer the question below; if you pick (i), fill in the blank with the percent. Then, explain your choice.

The percent of surveyed U.S. adults who had a great deal of confidence in both scientists and religious leaders

(i) is equal to ________________________________.

(ii) cannot be found with the information in the article.

(ii) cannot be found with the information in the article. Might be any value from 0-9. The reference material gives a bunch of rules when calculating possibilities, while the question asks about proportions which those rules cannot be applied to.

Question 5d) (This part is a continuation of the previous two.) Pick all of the options (i)-(iv) that are true for all values of p. Explain by algebraic or probailistic reasoning; you are welcome to use your function `no_disease_given_negative` to try a few cases numerically. Your explanation should include the reasons why you *didn't* choose some options.

$P(N \mid T_N)$ is

(i) equal to 0.95.

(ii) equal to 0.999×0.95.

(iii) greater than 0.999×0.95.

(iv) greater than 0.95.

Pick (iii) and (iv).

Explain by algebra as following: $\because (1-p)$ monotonously decreases
$\therefore (0.001(1-p) + 0.999 \times 0.95)$ monotoously decreases
$\therefore (0.999 \times 0.95)/(0.001(1-p) + 0.999 \times 0.95)$ monotoously increases
$\because$ no_disease_given_negative(0)$> 0.95 > 0.999 \times 0.95$
$\therefore$ Pick (iii) and (iv) rather than (i) and (ii)

Explain by probability as following: $P(N \mid T_N) = \frac{P(N)P(T_N|N)}{P(T_N)}$
$\because \frac{P(N)}{P(T_N)} > 1$
$\therefore P(N \mid T_N) > P(T_N \mid N) = 0.95$

Question 5e) Suzuki is one of most commonly owned makes of cars in our county (Alameda). A car heading from Berkeley to San Francisco is pulled over on the freeway for speeding. Suppose I tell you that the car is either a Suzuki or a Lamborghini, and you have to guess which of the two is more likely.

What would you guess, and why? Make some reasonable assumptions and explain them (data scientists often have to do this), justify your answer, and say how it's connected to the previous parts.

$P(Suzuki \mid Speeding) = P(Suzuki)P(Speeding \mid Suzuki)$

$P(Lamborghini \mid Speeding) = P(Lamborghini)P(Speeding \mid Lamborghini)$

$\because$ The question tells us $P(Suzuki) > P(Lamborghini)$ but fails to provide information on $P(Speeding \mid Suzuki)$ and $P(Speeding \mid Lamborghini)$

$\therefore$ A scientific guess cannot be provided with information given

To give reasonable assumptions that $P(Speeding \mid Suzuki) << P(Speeding \mid Lamborghini)$, one may draw the conclusion that a Lamborghini is more likely in such a senario. Still, the assumption itself is not so convincing as to what extent it happens, rendering the guess not so convincing too.

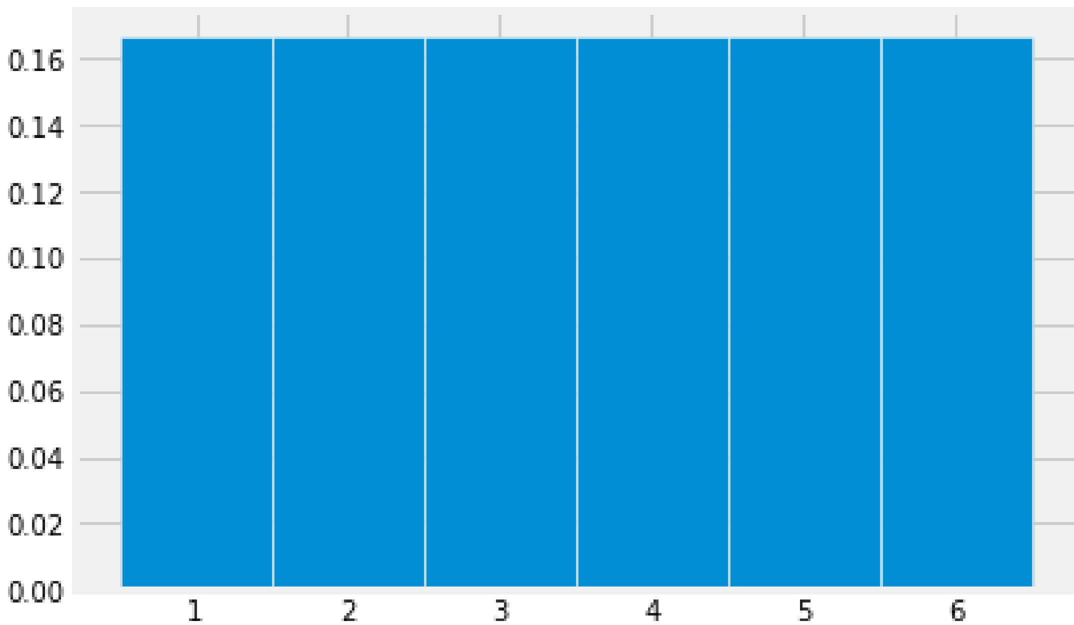
The connection to previous parts is obvious in the way by which the writer analyzes the probablity.

Question 6a) Define a function `integer_distribution` that takes an array of integers and draws the histogram of the distribution using unit bins centered at the integers and white edges for the bars. The histogram should be drawn to the density scale. The left-most bar should be centered at the smallest integer in the array, and the right-most bar at the largest.

Your function does not have to check that the input is an array consisting only of integers. The display does not need to include the printed proportions and bins.

If you have trouble defining the function, go back and carefully read all the lines of code that resulted in the probability histogram of the number of spots on one roll of a die. Pay special attention to the bins.

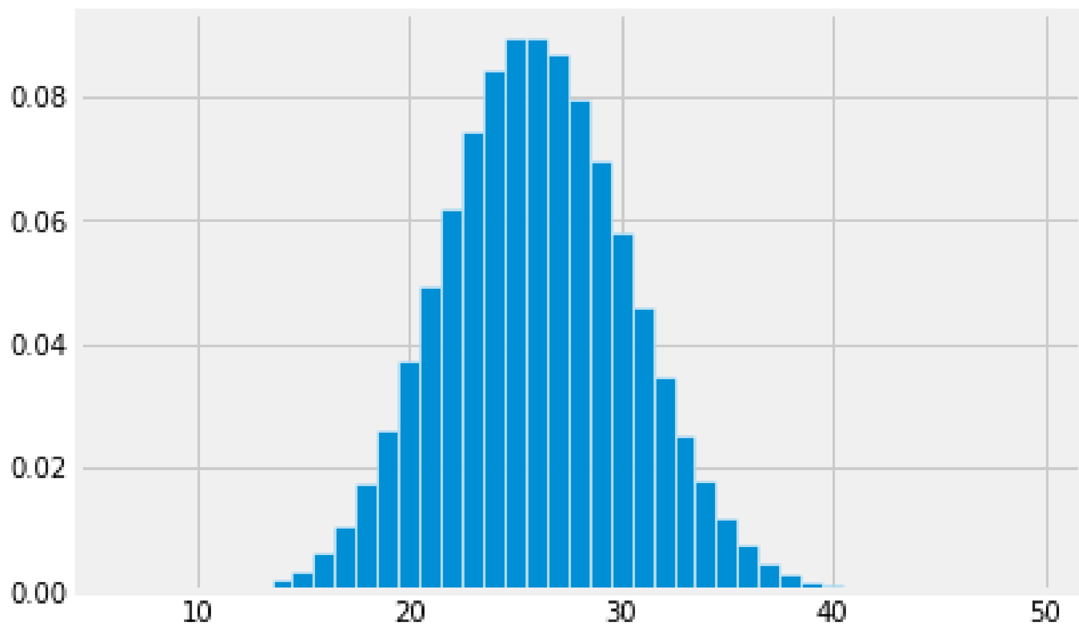
In [106]:
```
def integer_distribution(x):
    unit_bins = np.arange(min(x)-0.5, max(x)+0.6)
    plt.hist(x, bins = unit_bins, ec='white', density=True)
integer_distribution(faces)
```



Question 6c) Replace the "..." in the code cell below with a Python expression so that the output of the cell is an empirical histogram of 500,000 simulated counts of black people in 100 draws made at random with replacement from the population eligible for Swain's jury panel.

After you have drawn the histogram, you might want to take a moment to recall the conclusion reached in Data 8 based on the data that Swain's panel had 8 black people in it.

```
In [115]: simulated_counts = np.random.multinomial(100, [0.26, 0.74], size=500000)[:,0]
          integer_distribution(simulated_counts)
```



Question 6d) As you know, the count of black people in a sample of 100 people drawn at random from the eligible population is expected to be 26. Just by looking at the histogram in Part **c**, and **no other calculation**, pick the correct option and explain your choice. You might want to refer to the Data 8 textbook again.

The SD of the distribution of the number of black people in a random sample of 100 people drawn from the eligible population is closest to

(i) 1.4

(ii) 4.4

(iii) 7.4

(iv) 10.4

(ii)

Reason: average $\pm$ 3 SDs should include almost everything. (39-13)/6=4.33

Question 6e) The *normal curve with mean* μ *and SD* σ is defined by

$$f(x) \;=\; \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \qquad -\infty < x < \infty$$

Redraw your histogram from Part **c** and overlay the normal curve with$\mu = 26$andσequal to the choice you made in Part **d**. You just have to call `plt.plot` after `integer_distribution`. Use `np.e` for$e$. For the curve, use 2 as the line width, and any color that is easy to see over the blue histogram. It's fine to just let Python use its default color.

Now you can see why centering the histogram bars over the integers was a good idea. The normal curve peaks at 26, which is the center of the corresponding bar.

In [120]:
```
mu = 26
sigma = 4.4
x = np.linspace(0, 50, 200)
f_x = 1/np.sqrt(2*np.pi)/sigma*np.e**(-1/2*((x-mu)/sigma)**2)
integer_distribution(simulated_counts)
plt.plot(x,f_x,lw=2)
```

Out[120]:
```
[<matplotlib.lines.Line2D at 0x7f8ecb7565f8>]
```