Week #5: Histograms, Functions

Data 8 Tutoring

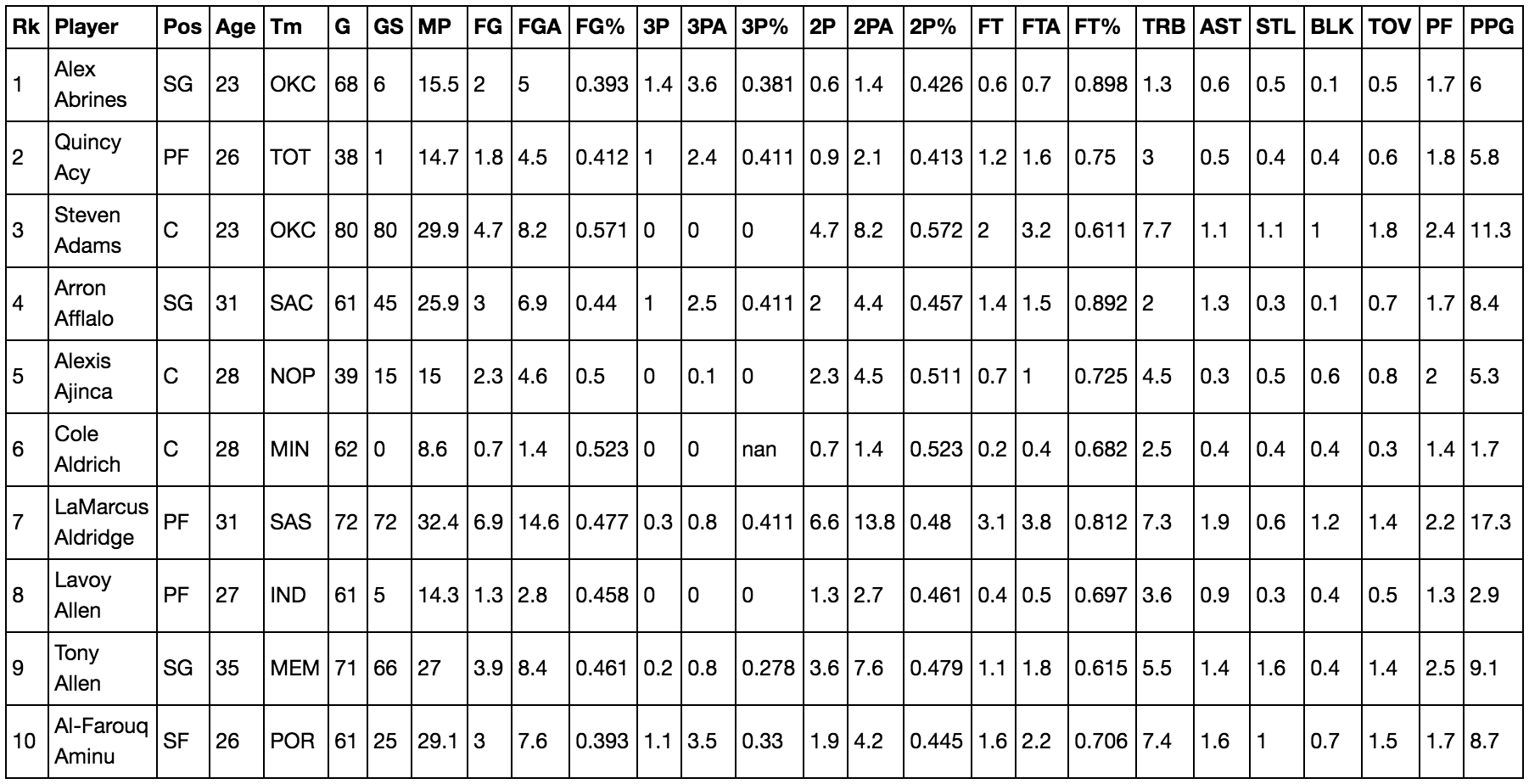
# 1 Histograms

## Key Concepts

* Histograms are used to visualize the distribution of numerical data.
* We use bins to group numerical variables into intervals. They are inclusive of their lower bounds but exclusive of their upper bounds, which is often expressed as [lower, upper).
  + Bins need not always be the same size, so watch out for bins of unequal widths.
* The x-axis is in the units of the numerical variable that we are investigating.
* The y-axis, formally known as the density scale, measures the percent of data in the bin relative to the amount of space in the bin, usually denoted as percent per <unit of x-axis>.
  + The reason we refer to it as density is that the y-axis can tell us how tightly clustered the data is in the bin.
* The area of a bin is equal to the percentage of data in the bin.
  + The larger the area of the bin, the more data lies in the bin!

## Practice Problems

Throughout this first section, we’ll be focusing on the following table: nba.csv. It describes the average statistics of NBA players for the 2016-2017 season. Ask your tutor if anything seems confusing.  
  
The first few rows of the nba table look like this. There is one row for each player.



Assume all imports are correctly made.

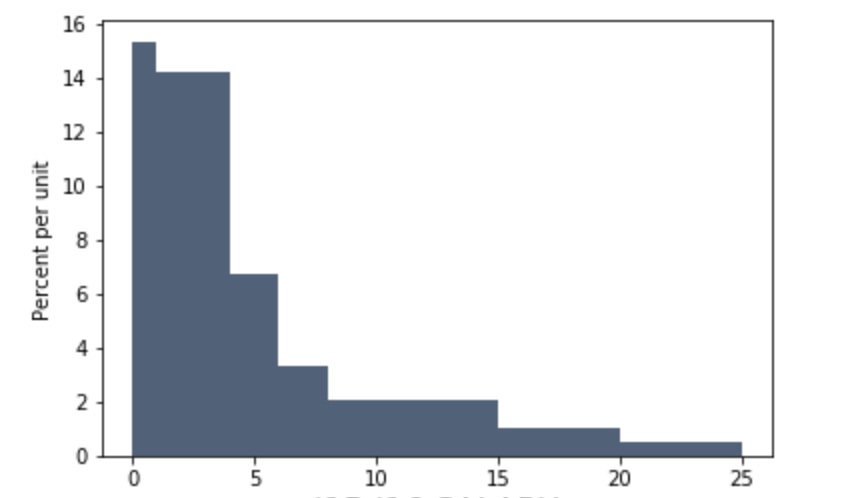
**1.1** NBA players must be at least 19 years old to play on a team. The oldest player last season was 40 years old. Create age\_bins and assign it to an array of equally spaced bin values that describe the ages of NBA players with a bin width of 2.

age\_bins = np.arange(19, 42, 2)

age\_bins

**1.2** Write code to create a histogram of the ages using the age\_bins you just created.

nba.hist('Age', bins=age\_bins)



Salary for players (millions)

**2.1** Let’s now view the above histogram generated from the nba\_salaries.csv table with the following code: nba\_salaries.hist(3,bins=make\_array(0,1,4,6,8,15,20,25)). Assume that all the players are represented in the histogram, and that the units for the salary data are in millions of dollars. Also note that this dataset contains 417 NBA Players. Answer the following questions with an arithmetic expression, or “Cannot answer”.

1. What percentage of players in the dataset make between zero and one million dollars? What percentage of players make between one and four million dollars? Which bin has more players?

((1-0)\*15) = 15 percent of players in the dataset make between zero and one million dollars

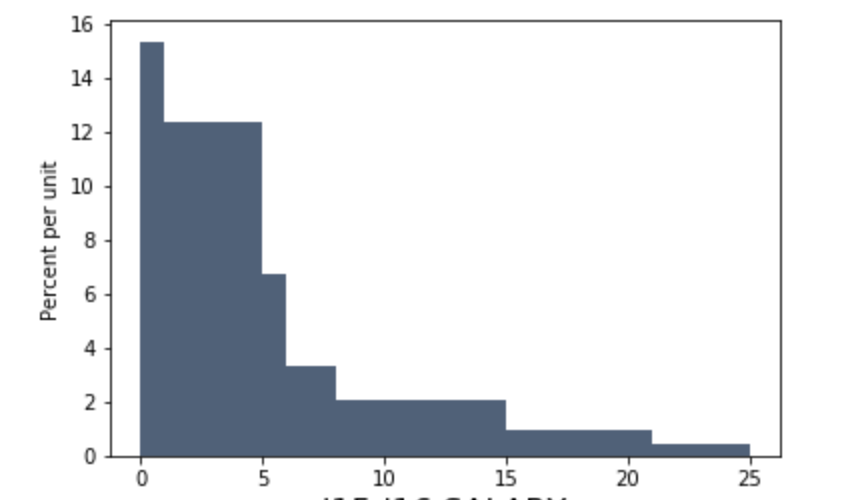
(14\*(4-1)) = 42 percent of players in the dataset make between one and four million dollars.

Since the bin from 1-4 has a greater percentage of players, the bin 1-4 has a greater number of players.

1. How many players make between 5 million and 6 million dollars?

Cannot find; we have no information about the 5 million to 6 million region, since we don’t know the distribution within the bin.

**2.2** Assume we have this second histogram generated using different bins: nba\_salaries.hist(3, bins=make\_array(0,1,5,6,8,15,21,25))



Salary for players (millions)

If you wrote “Cannot answer” for anything above, are you able to answer it now?

Yes, since we now have a histogram made on the same data with a 5-6 million bin.

Additionally, we would be able to find the number of players who make between 4 and 5 million, by

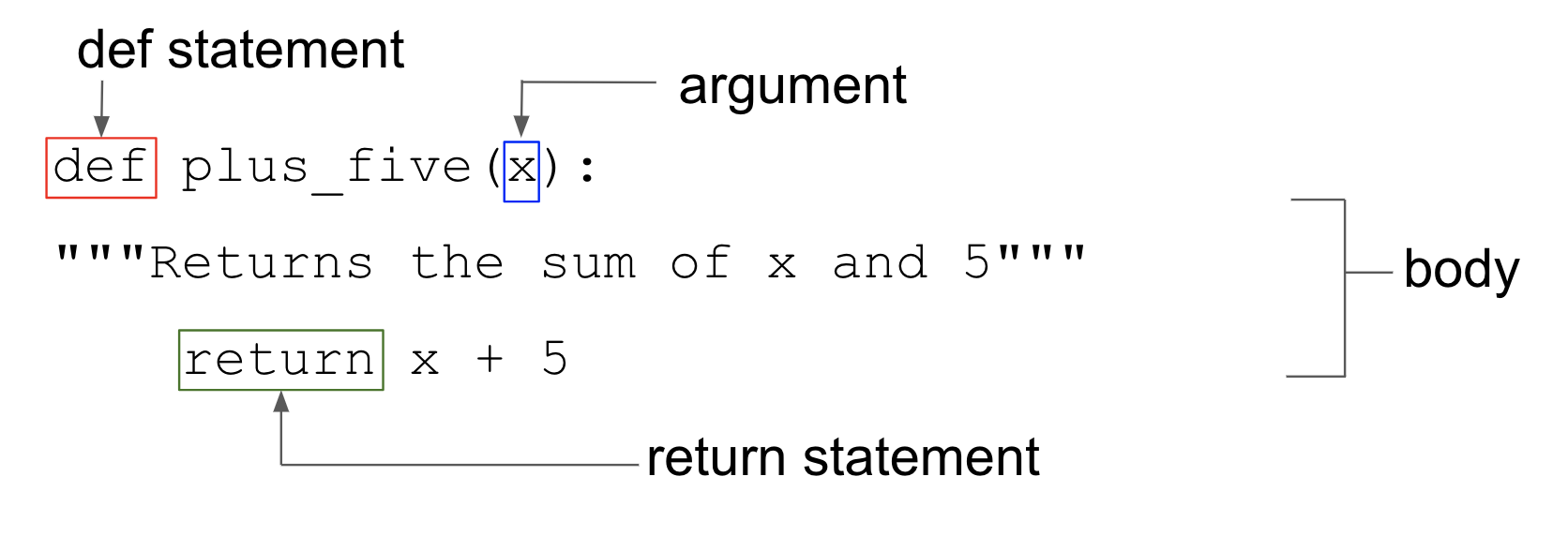
taking the number of players who make between 4 and 6 million and subtracting away the number of

players who make between 5 and 6 million.

# 2 Functions

## Key Concepts

Next, we will explore a tool that has been used many times already in this course: functions. We can define our own functions in order to give a name to a computational process that may be applied multiple times. The basic structure for a function is below:



## Practice Problems

**2.1** Define a function called calculate\_mean that takes in an array of numbers and returns the average of the numbers in the array. Don’t use the np.mean function!

def calculate\_mean(array):

sum\_of\_array = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

num\_elements = \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

return \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

def calculate\_mean(array):

sum\_of\_array = sum(array)

num\_elements = len(array)

return sum\_of\_array / num\_elements

**2.2** We have defined the function calculate\_statistics below. Analyze the function and decipher what it does, then answer the questions below.

def calculate\_statistics(array, multiplier):

largest\_num = max(array) (1)

smallest\_num = min(array) (2)

array\_average = calculate\_mean(array) (3)

stats\_array = make\_array(largest\_num, (4)

smallest\_num,

array\_average)

final\_array = stats\_array\*multiplier (5)

return final\_array (6)

Suppose you execute the line of code

statistics = calculate\_statistics(make\_array(5, 10, 15, 20), 2)

in a blank cell after defining the functions above. Answer the questions below.

a. After this function is called, what does largest\_num get assigned to?

largest\_num gets assigned to 20.

b. What does array\_average get assigned to?

array\_average gets assigned to 50/4, 12.5.

c. What does stats\_array get assigned to?

stats\_array gets assigned to an array that looks like [20, 5, 12.5].

d. What does final\_array get assigned to?

final\_array gets assigned to an array that looks like [40, 10, 25].

e. What does the function return? What is its type? (i.e. int, string)

The function returns an array with a value [40, 10, 25]. It is an array.

f. After the line of code is executed, what would happen if we tried to display the value of largest\_num?

It would tell us that largest\_num is not defined. The variable largest\_num is only defined within the scope of the function, so after the code has been executed, largest\_num no longer has any value.

g. Finally, if we ran calculate\_mean(statistics)after statistics is assigned, what would we get back as our output?

We would get the average of the statistics array! In this case, it is 25.