DATA 8

Fall 2020

Tutoring Section 5

Histograms *(Continued)* and Functional Programming

Logistics

- Tables Review: **Tabular Thinking** Guide
  - Link: http://data8.org/fa20/materials.html
- Much appreciated if you all could give some **feedback**:
    - Form: https://tinyurl.com/feedbackD8Kevin
- Tutor Office Hours (exclusively open for you all)
  - **Tuesday**: **10:30-11:00am** & **1:00-1:30pm**
  - *Please let me know if you are attending*
  - Questions/Concerns about literally anything
    - Life, college, hw, labs, discussion, tutoring sections, lecture
  - Same zoom link as tutoring sections!

All resources can be found on kevin-miao.com

Today

- Weekly Check-In
- Histograms
  - *Last Week:* Review
  - Practice Questions
  - Exam Question
- Functions
  - Quick Review
  - Practice Questions

Histograms

Last Week

- **When to use a histogram?**
  - Visualizing a distribution of numerical data
  - Mean/Median
- **Histograms**
  - **Areas** as **percentages**
  - **Height** as **densities**
  - The complete area under a histogram is always 1
  - Bins (can be arbitrary)
  - Formulas:

$$height = \frac{\%\ in\ a\ bin}{width\ of\ the\ bin}$$

$$area = \%\ = width\ of\ bin\ * height\ of\ bar$$

Worksheet

Link: **https://tinyurl.com/d8tutweek5**

Q1.1-1.2

1.1 NBA players must be at least 19 years old to play on a team. The oldest player that 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.

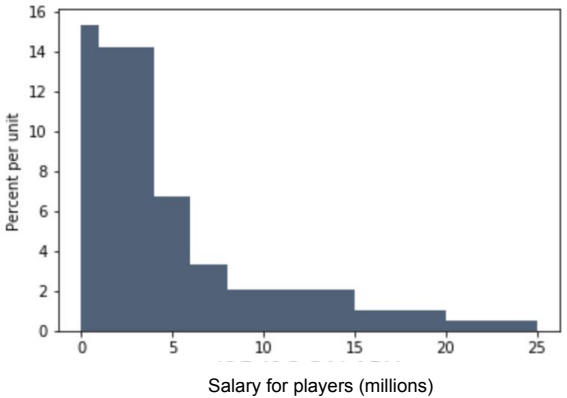
1.2 Write code to create a histogram of the ages using the `age_bins` you just created.

The first few rows of the `nba` table look like this. There is one row for each player.

Rk	Player	Pos	Age	Tm	G	GS	MP	FG	FGA	FG%	3P	3PA	3P%	2P	2PA	2P%	FT	FTA	FT%	TRB	AST	STL	BLK	TOV	PF	PPG
1	Alex Abrines	SG	23	OKC	68	6	15.5	2	5	0.393	1.4	3.6	0.381	0.6	1.4	0.426	0.6	0.7	0.898	1.3	0.6	0.5	0.1	0.5	1.7	6
2	Quincy	PF	26	TOT	38	1	14.7	1.8	4.5	0.412	1	2.4	0.411	0.9	2.1	0.413	1.2	1.6	0.75	3	0.5	0.4	0.4	0.6	1.8	5.8

Q2.1

2.1 Let's now view the histogram below 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". If you cannot answer the question, explain why.



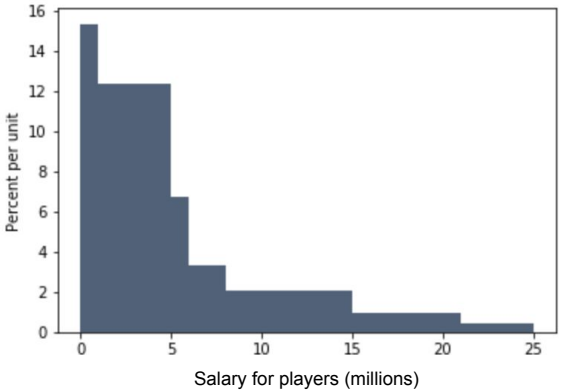
a. What percentage of players in the dataset make between zero and one million dollars? What percentage of players make between one and four million dollars? Which bin has more players?

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

Q2.2

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))
```



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

Functional Programming

- **What is a *function*?**
  - Analogous: $f(x) = 2x + 1$
  - Blackbox implementation
    - Feed in ***arguments***
    - Evaluates to a ***value***

def statement

- ```
  def cm_to_m(cm):
      """Converts centimeters to meters"""
      m = cm / 100
      return m
  ```

body

return statement
  ```

Q3.1

3.1 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 _____________________________________________________________
```

Q3.2 (abcd)

3.2 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 below in a blank cell. Answer the questions below.

```
statistics = calculate_statistics(make_array(5, 10, 15, 20), 2)
```

What does each of the following get assigned to?

1. `largest_num`
2. `array_average`
3. `stats_array`
4. `final_array`

Q3.2 (efg)

3.2 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 below in a blank cell. Answer the questions below.

```
statistics = calculate_statistics(make_array(5, 10, 15, 20), 2)
```

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

After the line is executed, what would happen to the value of `largest_num`?

What happens if we run `calculate_mean(statistics)`? **(from Q3.1)**

End of Section

- Please complete the anonymous Feedback form so I can improve my teaching:
  - **https://tinyurl.com/feedbackD8Kevin**

- Solutions and notes will be posted as soon as possible.