

**Data Science**



Workbook v0.9b

Brought to you by the Bootstrap team:

* Emmanuel Schanzer
* Kathi Fisler
* Shriram Krishnamurthi
* Sam Dooman
* Ed Campos

Bootstrap is licensed under a Creative Commons 3.0 Unported License. Based on a work from www.BootstrapWorld.org. Permissions beyond the scope of this license may be available at schanzer@BootstrapWorld.org.

**Unit 1**

*The question I chose is…*

|  |
| --- |
|  |
|  |
|  |
|  |
| *I think the answer is…*   |  | | --- | |  | |  | |  | |  | | *To find the answer, I would need to know…*   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | |  | | |

# Numbers and Strings

Make sure you’ve loaded the Unit 1 Starter File, and clicked “Run”.

1. Try typing 42 into the Interactions Area and hitting “Enter”. What happens?
2. Try typing in other Numbers. What happens if you try a decimal like 0.5? A fraction like 1/3? Try really big Numbers, and really small ones.
3. String values are always in quotes. Try typing your name in quotes, and see what happens when you hit “Enter”.
4. Try typing your name *without* the closing quote. What happens? Now try typing it without *any* quotes.
5. Is 42 the same as “42”? Why or why not? Write your answer below:
6. Just like in math, Pyret has *operators* like + and -. Try typing in 4 + 2, and then 4+2 (without the spaces). What can you conclude from this? Write your answer below:
7. Try typing in 4 + 2 + 6, 4 + 2 \* 6, and 4 + (2 \* 6). What can you conclude from this? Write your answer below:
8. Try typing in 4 + “cat”, and then “dog” + “cat”. What can you conclude from this? Write your answer below:

# Booleans

Boolean expressions are yes-or-no questions, and you probably already know some Boolean operators from math class, which compare Numbers. What do you think each of the following expressions will evaluate to? Try typing some into Pyret to experiment.

|  |  |
| --- | --- |
| 3 <= 4 \_\_\_\_\_\_\_\_\_\_\_  3 == 2 \_\_\_\_\_\_\_\_\_\_\_  2 <> 4 \_\_\_\_\_\_\_\_\_\_\_  3 <> 3 \_\_\_\_\_\_\_\_\_\_\_ | “a” > “b” \_\_\_\_\_\_\_\_\_\_\_  “a” <> “b” \_\_\_\_\_\_\_\_\_\_\_  “a” == “b” \_\_\_\_\_\_\_\_\_\_\_  “a” <> “a” \_\_\_\_\_\_\_\_\_\_\_ |

1. How many Number values are there? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. How many String values are there? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. How many Boolean values are there? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Boolean Operators

Pyret also has operators that work on *Booleans*. For each expression below, *write down your guess* about what it will evaluate to. Then type them in and see if you were right!

# (3 <= 4) and (3 == 2) \_\_\_\_\_\_\_\_\_\_\_\_

# (“a” == “b”) and (3 <> 4) \_\_\_\_\_\_\_\_\_\_\_\_

# (3 <= 4) or (3 == 2) \_\_\_\_\_\_\_\_\_\_\_\_

# (“a” == “b”) or (3 <> 4) \_\_\_\_\_\_\_\_\_\_\_\_

# **Unit 2**

An animal shelter has a database of all their animals. Here’s a small sample, showing just a few of those animals.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Name** | **Animal** | **Age (years)** | **Fixed** | **Legs** | **Weight (lbs)** | **Weeks-to-Adoption** |
| Felix | cat | 16 | TRUE | 4 | 9.2 | 5 |
| Mr. Peanutbutter | dog | 10 | FALSE | 4 | 161 | 22 |
| Nori | dog | 6 | TRUE | 4 | 35.3 | 2 |
| Nibblet | rabbit | 6 | FALSE | 4 | 4.3 | 2 |
| Fritz | dog | 4 | TRUE | 4 | 92 | 6 |
| Mittens | cat | 2 | TRUE | 4 | 7.4 | 5 |
| Lucky | dog | 3 | FALSE | 3 | 45.4 | 9 |
| Kujo | dog | 8 | FALSE | 4 | 172 | 20 |
| Buddy | lizard | 2 | FALSE | 4 | 0.3 | 12 |
| Boo-boo | dog | 11 | TRUE | 4 | 123 | 24 |
| Maple | dog | 3 | TRUE | 4 | 51.6 | 9 |
| Snowcone | cat | 2 | TRUE | 4 | 6.1 | 5 |
| Wade | cat | 1 | FALSE | 4 | 3.2 | 4 |
| Snuggles | tarantula | 2 | FALSE | 8 | 0.1 | 1 |

1. Put a dot next to all the animals that are cats.
2. Circle all the ages older than 4.
3. Put a number next to each row, ranking the animals by weight from lightest-to-heaviest.
4. Cross out all the animals that are neither dogs nor cats.

# The Design Recipe

**Define a function called birth-year, which calculates the year an animal was born:**

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_  *(animal :: Row)*  *Number*  *Number(animal :: Row)*  *birth-year*  name domain range  *Consumes an animal, subtracts age from the current year to produce the birth-year born*  # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| **examples:**  *2018 - pet1[“age”]*  *pet1*  *birth-year*  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |
| **fun** \_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_) : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |

**Define a function called is-kitten, which consumes a Row of the animals table and produces true if it's a cat less than 2 years old.**

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_  name domain range  # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| **examples:**  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |
| **fun** \_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_) : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |

**Define a function called nametag, prints out each animal's name in big red letters.**

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_  name domain range  # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| **examples:**  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |
| **fun** \_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_) : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |

**Define a function called is-fixed, which consumes a Row of the animals table and produces true if it's an animal that's been fixed.**

|  |  |
| --- | --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_  name domain range  # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | |
| **examples:**  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |
| **fun** \_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_) : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |

**Define a function called sentence, which consumes a Row of the animals table and produces a String containing the animal's name, the string " the ", and the species of the animal. (For example, "Nori the dog").**

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_  name domain range  # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **examples:**  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |
| **fun** \_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_) :  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |

**What kind of animal would *you* adopt? Write a function called adopt, which consumes a Row of the animals table and produces true if it's an animal that you would adopt.**

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_  name domain range  # \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **examples:**  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_) **is**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |
| **fun** \_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_) :  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  **end** |

# My Dataset

1. My dataset is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Some of the columns in my dataset are :

|  |  |  |
| --- | --- | --- |
| **Name** (capitalization matters !) | **Datatype** | **Quantitative/Categorical** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Three questions I have about my dataset:**

**1.**

**2.**

**3.**

# **Unit 3**

1. How many functions are defined in this file? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. What is the name of the last function? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. What is the Domain of the last function? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. What is the Range of the last function? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. What is the Range of the last function? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. What is the variable name that the last function uses? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
7. Which function will tell us if an animal is a kitten? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
8. Which function will print out “<name> the <species>”? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
9. Which function will tell us if an animal is a dog older than 10? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
10. Which function will tell us if an animal has been fixed? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
11. Which function will draw a nametag for an animal? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
12. One of the examples for the last function is broken. Fix this example in the Definitions Area.

# Table Plan

On Kitten Day, the shelter prints up a list of all the cats in their database that are less than 2 years old, and makes nametags for them. They need a function that will help them out! Define a function called get-kittens-tags, which takes in the dataset and produces the correct table.

|  |
| --- |
| **Contract and Purpose**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Table*  *(animals :: Table)*  *get-kittens-tags*    *# Consume a table of animals, and produce a table containing kittens with nametags, sorted by name*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Example Tables**  Make a Start Table and a result based on that table.   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **name** | **species** | **age** | **fixed** | **legs** | **weight** | **adopt** | **tag** | | Sasha | cat | 1 | FALSE | 4 | 6.5 | 4 | Sascha | | Wade | cat | 1 | FALSE | 4 | 3.2 | 4 | Wade |   animals-table 🡪 get-kittens-tags(animals-table)   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **name** | **species** | **age** | **fixed** | **legs** | **weight** | **adopt** | | Sasha | cat | 1 | FALSE | 4 | 6.5 | 4 | | Toggle | dog | 3 | TRUE | 4 | 48 | 3 | | Buddy | lizard | 2 | FALSE | 4 | 0.3 | 12 | | Wade | cat | 1 | FALSE | 4 | 3.2 | 4 | | Mittens | cat | 2 | TRUE | 4 | 7.4 | 5 | |
| **Define the function**  Use the relevant methods (circle your helper functions!), then produce a result with the new table.  **fun** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_):  *get-kittens-tags*  *pets*  *Define the table*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *t = pets*  *.build-columns( )*  *.filter( )*  *.order-by( )*  *t*  *Are there more columns? Are there fewer rows? Are the rows ordered?*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Produce the result*  **end** |

# Table Plan

The first weekend of every month, the shelter holds a “meet the dogs” picnic, to encourage families to adopt their dogs. Write a function called get-dogs-by-age, that takes their database and produces a table of all the dogs in the shelter, sorted from youngest to oldest.

|  |
| --- |
| **Contract and Purpose**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Table*  *(animals :: Table)*  *get-dogs-by-age*    *# Consume a table of animals, and produce a table containing only the dogs, sorted by age*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Examples**   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **name** | **species** | **age** | **fixed** | **legs** | **weight** | **adopt** | | Toggle | dog | 3 | TRUE | 4 | 48 | 3 | | Fritz | dog | 4 | TRUE | 4 | 92 | 6 |   Make a Start Table and a result based on that table.  animals-table 🡪 get-dog-by-age(animals-table)   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **name** | **species** | **age** | **fixed** | **legs** | **weight** | **adopt** | | Snowcone | cat | 2 | TRUE | 4 | 6.1 | 5 | | Wade | cat | 1 | FALSE | 4 | 3.2 | 4 | | Hercules | cat | 3 | FALSE | 4 | 13.4 | 7 | | Toggle | dog | 3 | TRUE | 4 | 48 | 3 | | Fritz | dog | 4 | TRUE | 4 | 92 | 6 | |
| **Define the function**  Use the relevant methods (circle your helper functions!), then produce a result with the new table.  **fun** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_):  *Define the table*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *.build-columns( )*  *.filter( )*  *.order-by( )*  *Are there more columns? Are there fewer rows? Are the rows ordered?*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Produce the result*  **end** |

# Table Plan

It’s important for animals to stay healthy, especially when they get older. The veterinarians at the shelter want to put some of the dogs on a diet! They need a regular report of all the older dogs, sorted from heaviest-to-lightest. Define a function old-dogs-diet, which does just that!

|  |
| --- |
| **Contract and Purpose**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Examples**  Make a Start Table and a result based on that table.   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **name** | **species** | **age** | **fixed** | **legs** | **weight** | **adopt** | | Lucky | dog | 3 | TRUE | 3 | 45.4 | 9 | | Snowcone | cat | 2 | TRUE | 4 | 6.1 | 5 | | Toggle | dog | 3 | TRUE | 4 | 48 | 3 |   animals-table 🡪 get-fixed-by-legs(animals-table)   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **name** | **species** | **age** | **fixed** | **legs** | **weight** | **adopt** | | Snowcone | cat | 2 | TRUE | 4 | 6.1 | 5 | | Lucky | dog | 3 | TRUE | 3 | 45.4 | 9 | | Hercules | cat | 3 | FALSE | 4 | 13.4 | 7 | | Toggle | dog | 3 | TRUE | 4 | 48 | 3 | | Snuggles | tarantula | 2 | FALSE | 8 | 0.1 | 1 | |
| **Define the function**  Use the relevant methods (circle your helper functions!), then produce a result with the new table.  **fun** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_):  *Define the table*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *t =*  *.build-columns( )*  *.filter( )*  *.order-by( )*  *Are there more columns? Are there fewer rows? Are the rows ordered?*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Produce the result*  **end** |

# Table Plan

The shelter is tracking birth-years for all the animals who’ve been fixed. They need a function that takes in their database and returns a table that contains the birth-year for each one. Define get-fixed-birth that will do this for them.

|  |
| --- |
| **Contract and Purpose**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Examples**  Make a Start Table and a result based on that table.   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | **name** | **species** | **age** | **fixed** | **legs** | **weight** | **adopt** | **year** | | Snowcone | cat | 2 | TRUE | 4 | 6.1 | 5 | 2015 | | Lucky | dog | 3 | TRUE | 3 | 45.4 | 9 | 2014 | | Toggle | dog | 3 | TRUE | 4 | 48 | 3 | 2014 |   animals-table 🡪 get-fixed-by-legs(animals-table)   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **name** | **species** | **age** | **fixed** | **legs** | **weight** | **adopt** | | Snowcone | cat | 2 | TRUE | 4 | 6.1 | 5 | | Lucky | dog | 3 | TRUE | 3 | 45.4 | 9 | | Hercules | cat | 3 | FALSE | 4 | 13.4 | 7 | | Toggle | dog | 3 | TRUE | 4 | 48 | 3 | | Snuggles | tarantula | 2 | FALSE | 8 | 0.1 | 1 | |
| **Define the function**  Use the relevant methods (circle your helper functions!), then produce a result with the new table.  **fun** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_):  *Define the table*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *t =*  *.build-columns( )*  *.filter( )*  *.order-by( )*  *Are there more columns? Are there fewer rows? Are the rows ordered?*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Produce the result*  **end** |

# My Dataset

**Organizing my Dataset**

1. One way to *order* my dataset is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. A *subset* of my data might be \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. A useful column to compute is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Unit 4**

# Measuring Center

You have nine animals, whose weight (in pounds) are:

10, 10, 30, 35, 35, 35, 40, 60, 120

1. What is the mean weight of this list, in pounds? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. What is the median weight of this list, in pounds? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. What is/are the mode(s) of this list, in pounds? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Suppose we break this list into three smaller lists:

10, 10, 30

35, 35, 40

50, 60, 120

1. What is the median of the first list? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. What is the median of the second list? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. What is the median of the third list? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Is this different from the median of the whole list (#2)? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Table Plan

The shelter wants a function that will calculate the median age of all the dogs in the shelter. Write a function called median-dog-age that will take in a table of animals and do just that.

|  |
| --- |
| **Contract and Purpose**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Examples**  Make a Start Table and a result based on that table.  animals-table 🡪 median-dog-age(animals-table)   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **name** | **species** | **age** | **fixed** | **legs** | **weight** | **adopt** | | Snowcone | cat | 2 | TRUE | 4 | 6.1 | 5 | | Lucky | dog | 3 | TRUE | 3 | 45.4 | 9 | | Hercules | cat | 3 | FALSE | 4 | 13.4 | 7 | | Toggle | dog | 3 | TRUE | 4 | 48 | 3 | | Snuggles | tarantula | 2 | FALSE | 8 | 0.1 | 1 | |
| **Define the function**  Use the relevant methods (circle your helper functions!), then produce a result with the new table.  **fun** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_):  *Define the table*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *t =*  *.build-columns( )*  *.filter( )*  *.order-by( )*  *Are there more columns? Are there fewer rows? Are the rows ordered?*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Produce the result*  **end** |

# Table Plan

The shelter wants to know how long a kitten stays at the shelter before finding a “forever home”. Define a function called mean-kitten-adoption, that will calculate the mean of the length of time it takes for kittens to be adopted when given the dataset.

|  |
| --- |
| **Contract and Purpose**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Examples**  Make a Start Table and a result based on that table.  animals-table 🡪 median-dog-age(animals-table)   |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **name** | **species** | **age** | **fixed** | **legs** | **weight** | **adopt** | | Snowcone | cat | 2 | TRUE | 4 | 6.1 | 5 | | Lucky | dog | 3 | TRUE | 3 | 45.4 | 9 | | Hercules | cat | 3 | FALSE | 4 | 13.4 | 7 | | Toggle | dog | 3 | TRUE | 4 | 48 | 3 | | Snuggles | tarantula | 2 | FALSE | 8 | 0.1 | 1 | |
| **Define the function**  Use the relevant methods (circle your helper functions!), then produce a result with the new table.  **fun** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_):  *Define the table*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *t =*  *.build-columns( )*  *.filter( )*  *.order-by( )*  *Are there more columns? Are there fewer rows? Are the rows ordered?*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Produce the result*  **end** |

# My Dataset

**Measures of Center**

1. The column I choose to measure is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. The **mean** of that column is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. The **median** of that column is \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. The **mode(s)** of that column is/are \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Based on the differences between mean and median, I conclude : \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# **Unit 5**

Use the Table below to help you answer the questions.

|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **species** | **age** | **pounds** |
| Sasha | cat | 1 | 6.5 |
| Felix | cat | 16 | 9.2 |
| Wade | cat | 1 | 3.2 |
| Boo-boo | dog | 11 | 123 |
| Maple | dog | 3 | 51.6 |
| Nori | dog | 6 | 35.3 |
| Nibblet | rabbit | 6 | 4.3 |

1. Which animal(s) is the heaviest? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Which animal(s) is the youngest? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. How much of the *total weight* comes from Maple? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. How much of the *combined age* comes from Nori? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Would these questions be harder to answer if the table had 100 rows? If so, why?

# Visualizing Quantity

Use the charts below to help you answer the questions.

|  |  |
| --- | --- |
|  |  |
| Animals Ages (yrs) | Animals Weights (lbs) |

1. Which animal(s) is the heaviest? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Which animal(s) is the youngest? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. How much of the *total weight* comes from Maple? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. How much of the *combined age* comes from Nori? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Which chart did you use for questions 1 and 2? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. Which chart did you use for questions 3 and 4? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
7. What kind of questions are Bar Charts better for than Pie Charts?

# Table Plan

Dogs are generally a lot bigger heavier than cats, so the shelter wants to look at a chart of *only* the dogs to determine who needs more exercise time. Define a function pie-dog-weight, which will make a pie chart showing the relative weights of all the dogs in the shelter.

|  |
| --- |
| **Contract and Purpose**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Examples**  Make a Start Table and a result based on that table.  animals-table 🡪 pie-dog-weight(animals-table)   |  |  |  | | --- | --- | --- | | **name** | **…** | **weight** | | Snowcone | … | 6.1 | | Lucky | … | 45.4 | | Hercules | … | 13.4 | | Toggle | … | 48 | | Snuggles | … | 0.1 | |
| **Define the function**  Use the relevant methods (circle your helper functions!), then produce a result with the new table.  **fun** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_):  *Define the table*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *t =*  *Are there more columns? Are there fewer rows? Are the rows ordered?*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Produce the result*  **end** |

# Bad Starter Tables!

For each word problem, a Starter Table must have (1) all the columns that matter, (2) a representative sample of the rows, and be in (3) random order. For each of the problem below, check the boxes to determine if the Starter Table meets those criteria.

1. **The shelter wants to know the median age of all the cats**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **name** | **species** | **age** | **fixed** | **legs** | **pounds** | **weeks** |
| Sasha | cat | 1 | FALSE | 4 | 6.5 | 3 |
| Mittens | cat | 2 | TRUE | 4 | 7.4 | 5 |
| Sunfower | cat | 5 | TRUE | 4 | 8.1 | 10 |

* Relevant columns
* Representative sample of rows
* Random order

1. **The shelter wants a pie chart showing all the dogs’ weight**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **name** | **species** | **age** | **fixed** | **legs** | **pounds** | **weeks** |
| Sasha | cat | 1 | FALSE | 4 | 6.5 | 3 |
| Mittens | cat | 2 | TRUE | 4 | 7.4 | 5 |
| Sunfower | cat | 5 | TRUE | 4 | 8.1 | 10 |

|  |  |  |
| --- | --- | --- |
| **name** | **species** | **age** |
| Fritz | dog | 4 |
| Wade | cat | 2 |
| Nibblet | rabbit | 6 |
| Daisy | dog | 5 |

* Relevant columns
* Representative sample of rows
* Random order

1. **Sort all the animals alphabetically by name**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **name** | **species** | **age** | **fixed** | **legs** | **pounds** | **weeks** |
| Ada | dog | 2 | TRUE | 4 | 32 | 3 |
| Bo | dog | 4 | TRUE | 4 | 76.1 | 10 |
| Boo-boo | dog | 11 | TRUE | 4 | 123 | 10 |

* Relevant columns
* Representative sample of rows
* Random order

1. **Make a bar chart for all the fixed animals**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **name** | **species** | **age** | **fixed** | **legs** | **pounds** | **weeks** |
| Sasha | cat | 1 | FALSE | 4 | 6.5 | 3 |

* Relevant columns
* Representative sample of rows
* Random order

# Table Plan

Define a function bar-kitten-adoption, which takes in a Table of animals and creates a bar chart showing how many weeks it took for each kitten to be adopted

|  |
| --- |
| **Contract and Purpose**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Examples**  Make a Start Table and a result based on that table.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   |  | | --- | |  | |  | |  | |  | |  | |  | |
| **Define the function**  Use the relevant methods (circle your helper functions!), then produce a result with the new table.  **fun** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_):  *Define the table*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *t =*  *Are there more columns? Are there fewer rows? Are the rows ordered?*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Produce the result*  **end** |

# Table Plan

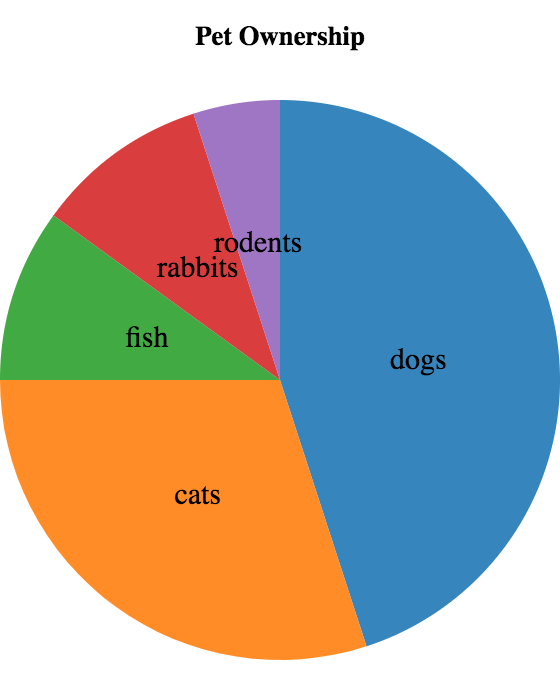
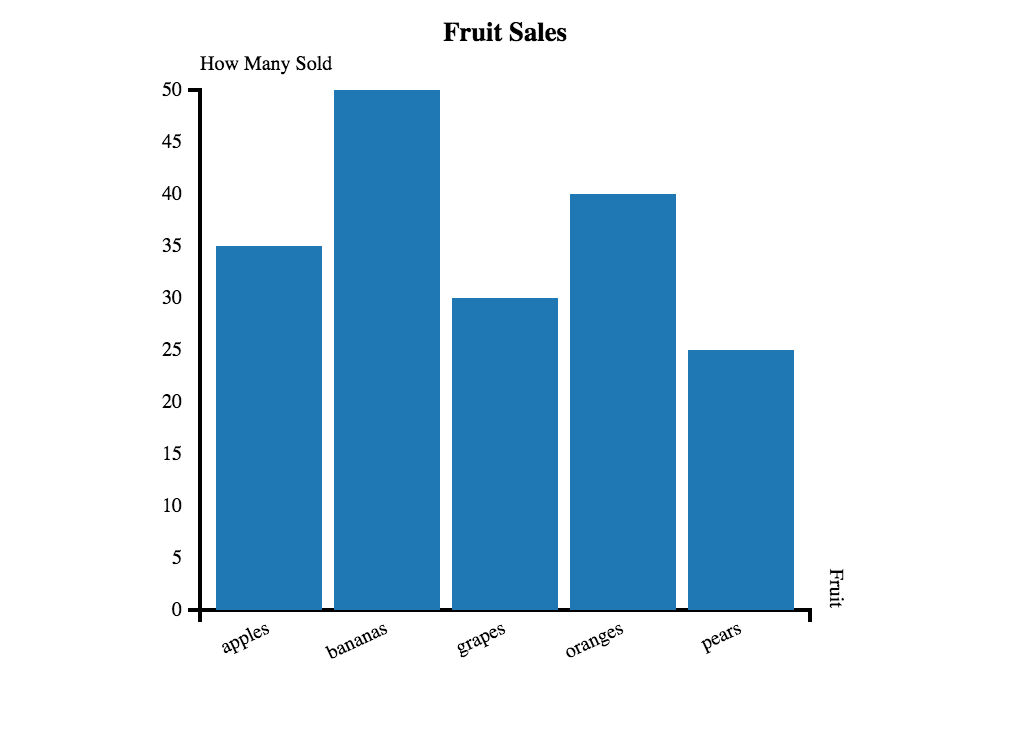
|  |
| --- |
| **Contract and Purpose**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_    \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| **Examples**  Make a Start Table and a result based on that table.  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 🡪 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_   |  | | --- | |  | |  | |  | |  | |  | |  | |
| **Define the function**  Use the relevant methods (circle your helper functions!), then produce a result with the new table.  **fun** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_):  *Define the table*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *t =*  *Are there more columns? Are there fewer rows? Are the rows ordered?*  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  *Produce the result*  **end** |

# My Dataset

**Visualizing Quantity**

# **Unit 6**

Use the two (unrelated) charts to answer the questions below…

1. Is the chart on the left a pie or bar chart? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Which pet is the least popular? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. How many pears were sold? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Which are more popular, fish or rodents? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Are apples more popular than grapes? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. How many categories of fruit are there? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
7. What fruit is least popular? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
8. Which pet is the most popular? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Visualizing Frequency

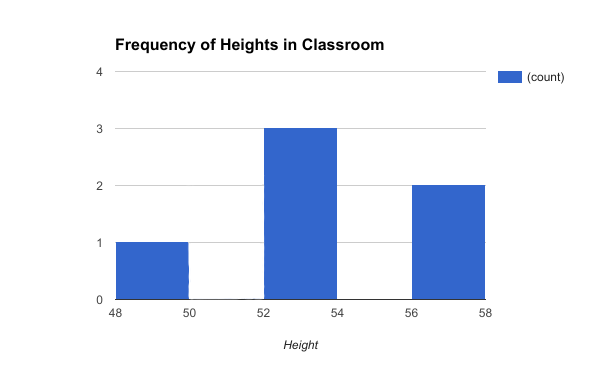
|  |  |  |  |
| --- | --- | --- | --- |
| **name** | **species** | **age** | **pounds** |
| "Sasha" | "cat" | 1 | 6.5 |
| "Boo-boo" | "dog" | 11 | 123 |
| "Felix" | "cat" | 16 | 9.2 |
| "Buddy" | "lizard" | 2 | 0.3 |
| "Nori" | "dog" | 6 | 35.3 |
| "Wade" | "cat" | 1 | 3.2 |
| "Nibblet" | "rabbit" | 6 | 4.3 |
| "Maple" | "dog" | 3 | 51.6 |

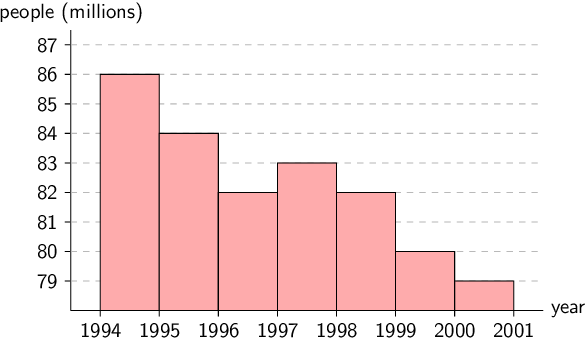
1. How many cats are there? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. How many dogs are there? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. How many animals are between 3-6 years old? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. How many weigh between 0-5 pounds? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
5. Are there more animals weighing 0-5 than 6-10 pounds? \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
6. The charts below are based on the Starter Table above. What is each one measuring? Write down your guess underneath each one.

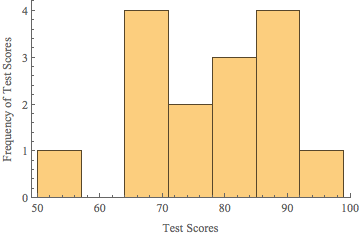
# Histograms

|  |  |  |
| --- | --- | --- |
| **First** | **Last** | **Height** |
| "John" | "Doe" | 52.0 |
| "Jane" | "Smith" | 49.1 |
| "Javon" | "Jackson" | 57.7 |
| "Angela" | "Enriquez" | 52.5 |
| "Jack" | "Thompson" | 53.0 |
| "Dominique" | "Rodriguez" | 51.1 |
| "Sammy" | "Carter" | 56.2 |
| "Andrea" | "Garcia" | 50.8 |

1. How many students are 48in or above, *and* less than 50in tall? \_\_\_\_\_\_\_\_\_\_\_\_
2. How many students are 50in or above, *and* less than 52in tall? \_\_\_\_\_\_\_\_\_\_\_\_
3. How many students are 52in or above, *and* less than 54in tall? \_\_\_\_\_\_\_\_\_\_\_\_
4. How many students are 54in or above, *and* less than 56in tall? \_\_\_\_\_\_\_\_\_\_\_\_
5. How many students are 56in or above, *and* less than 58in tall? \_\_\_\_\_\_\_\_\_\_\_\_
6. Add a bar to this histogram for students who are 50in or above, *and* less than 52in tall.

Histogram Practice

1. How many people were born between 1996 and 1997? \_\_\_\_\_\_\_\_\_\_\_
2. On what year were the most number of people born? \_\_\_\_\_\_\_\_\_\_\_
3. How many bins does this histogram have? \_\_\_\_\_\_\_\_\_\_\_\_
4. Were more people born in 1994 or 1995? \_\_\_\_\_\_\_\_\_\_\_\_



1. How many bins does this histogram have? \_\_\_\_\_\_\_\_\_\_\_\_
2. What is (are) the bins with the highest frequency of scores? \_\_\_\_\_\_\_\_\_\_\_\_
3. How many students scored between 85 and 92? \_\_\_\_\_\_\_\_\_\_\_\_

# **Unit 6**

***“The more you pay at a restaurant, the better it is.”***

*Do you agree?*

I hypothesize…

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

I found…

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

# Creating a Scatter Plot

For each row in the following table, add a dot to the scatter plot. The first 3 rows have been completed for you. Use the values from the left column along the horizontal axis, and values from the right column along the vertical axis.

|  |  |
| --- | --- |
| **Dollars** | **Fingers** |
| 0 | 3 |
| 1 | 5 |
| 2.5 | 1 |
| 2 | 2 |
| 6 | 0 |
| 4 | 3 |



# Grading Predictor Functions

Below are the scatterplots for 4 data sets, with two different predictors shown for each set. For each data set**, circle the plot with the predictor function that fits better**, and **give it a grade between 0 (worst possible fit) and 1 (best possible fit).**

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | wb-pred-a-1.png | wb-pred-a-2.png | Grade for best predictor:  \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 2 | wb-pred-b-2.png | wb-pred-b-1.png | Grade for best predictor:  \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 3 | wb-pred-c-2.png | wb-pred-c-1.png | Grade for best predictor:  \_\_\_\_\_\_\_\_\_\_\_\_\_ |
| 4 | wb-pred-d-2.png | wb-pred-d-1.png | Grade for best predictor:  \_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

# Checking for Understanding

|  |
| --- |
|  |

1. In your own words, explain what a **predictor function** is.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

1. In your own words, explain what the **r-squared** value of a predictor is.

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |

Contracts

|  |  |  |
| --- | --- | --- |
| **Name** | **Domain** | **Range** |
| triangle | :: (side :: *Number*, style :: *String*, color :: *String*) | 🡪 *Image* |
| circle | :: (radius :: *Number*, style :: *String*, color :: *String*) | 🡪 *Image* |
| star | :: (radius :: *Number*, style :: *String*, color :: *String*) | 🡪 *Image* |
| rectangle | :: (width :: *Num*, height :: *Num,* style :: *Str*, color :: *Str*) | 🡪 *Image* |
| ellipse | :: (width :: *Num*, height :: *Num,* style :: *Str*, color :: *Str*) | 🡪 *Image* |
| square | :: (size :: *Number*, style :: *String*, color :: *String*) | 🡪 *Image* |
| text | :: (str :: *String*, size :: *Number*, color :: *String*) | 🡪 *Image* |
| overlay | :: (img1 :: *Image*, img2 :: *Image*) | 🡪 *Image* |
| rotate | :: (degree :: *Number*, img :: *Image*) | 🡪 *Image* |
| scale | :: (factor :: *Number*, img :: *Image*) | 🡪 *Image* |
| string-repeat | :: (text :: *String*, repeat :: *Number*) | 🡪 *String* |
| num-sqr | :: (n :: *Number*) | 🡪 *Number* |
| num-sqrt | :: (n :: *Number*) | 🡪 *Number* |
| num-min | :: (a :: *Number, b:: Number*) | 🡪 *Number* |
| num-max | :: (a :: *Number, b:: Number*) | 🡪 *Number* |
| get-row | :: (t :: *Table, index :: Number*) | 🡪 *Row* |

Contracts

|  |  |  |
| --- | --- | --- |
| **Name** | **Domain** | **Range** |
| *<Table>.*row-n | :: (n :: *Number*) | 🡪 *Row* |
| *<Table>.*filter | :: (test :: *(Row 🡪 Boolean*) ) | 🡪 *Table* |
| *<Table>.*build-column | :: (col :: *String, builder* :: *(Row 🡪 Value)* ) | 🡪 *Table* |
| mean | :: (t:: *Table,* col :: *String*) | 🡪 *Number* |
| median | :: (t:: *Table,* col :: *String*) | 🡪 *Number* |
| modes | :: (t:: *Table,* col :: *String*) | 🡪 *List<Number>* |
| bar-chart | :: (t:: *Table,* labels :: *String*, values :: *String*) | 🡪 *Image* |
| pie-chart | :: (t:: *Table,* labels :: *String*, values :: *String*) | 🡪 *Image* |
| freq-bar-chart | :: (t:: *Table,* values :: *String*) | 🡪 *Image* |
| histogram | :: (t:: *Table,* values :: *String*, bin-width :: *Number*) | 🡪 *Image* |
| scatter-plot | :: (t:: *Table,* xs :: *String*, ys :: *String*) | 🡪 *Image* |
| labeled-scatter-plot | :: (t:: *Table,* xs :: *String*, ys :: *String*, labels :: *String*) | 🡪 *Image* |
|  | :: | 🡪 |
|  | :: | 🡪 |
|  | :: | 🡪 |
|  | :: | 🡪 |