





Workbook v1.0

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](http://www.BootstrapWorld.org). Permissions beyond the scope of this license may be available at [schanzer@BootstrapWorld.org](mailto:schanzer@BootstrapWorld.org).

# Unit 1

- Many important questions ("what's the best restaurant in town?", "is this law good for citizens?", etc.) are answered with data. Data Scientists try and answer these questions, by writing *programs that ask questions of data*.
- Data of all types can be organized into **Tables**
- Every Table has a **header row**, and some number of **data rows**
- **Quantitative data** is data - usually numeric - that measures *quantity*, such as a person's height, a score on test, a measure of distance, etc. A list of quantitative data can be ordered from smallest to largest.
- **Categorical data** is data that specifies *categories*, such as eye color, country of origin, etc. A list of categorical data has no notion of "smallest" or "largest", and cannot be ordered.
- **Programming languages** involves different *datatypes*, such as Numbers, Strings, Booleans and Images.
- **Operators** (like +, -, \*, <, etc.) are written between values. For example: `4 + 2`
- **Functions** (like triangle, star, string-repeat, etc.) are written first, followed by a list of **arguments** in parentheses. For example: `star(50, "solid", "red")`



# 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 will each of the shaded expressions below evaluate to? Type them into Pyret if you're not sure.

`3 <= 4`

\_\_\_\_\_

`3 == 2`

\_\_\_\_\_

`2 <> 4`

\_\_\_\_\_

`3 <> 3`

\_\_\_\_\_

`"a" > "b"`

\_\_\_\_\_

`"a" <> "b"`

\_\_\_\_\_

`"a" == "b"`

\_\_\_\_\_

`"a" <> "a"`

\_\_\_\_\_

---

## 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)`

\_\_\_\_\_

---

1. How many different Number values are there in Pyret? \_\_\_\_\_

2. How many different String values are there in Pyret? \_\_\_\_\_

3. How many different Boolean values are there in Pyret? \_\_\_\_\_

# Unit 2

- Programming languages let us **define our own function**.
- We use the **Design Recipe** to help us define functions without making mistakes.
- The first step is to write a **Contract** and **Purpose Statement** for the function, which specify the Name, Domain and Range of the function and give a summary of what it does.
- The second step is to **write at least two examples**, which show how the function should work for specific inputs. These examples help us see patterns, and we express those patterns by **circling and labeling** what changes.
- The final step is to **define the function**, which generalizes our examples.





# The Animals Dataset

1. My dataset is Animals from a pet store
2. Some of the columns in my dataset are :

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

3. Three questions I have about the animals dataset:

1.

---

---

2.

---

---

3.

---

---

# The Design Recipe

Define a function called `is-fixed`, which tells us whether or not an animal is fixed

|  |   |                         |   |                |
|--|---|-------------------------|---|----------------|
| <i>is-fixed</i>  | : | ( <i>animal</i> :: Row) | → | <i>Boolean</i> |
| name   |   | domain                  |   | range          |
| # Consumes an animal, and produces the value in the fixed column |   |                         |   |                |

**examples:**

```
    is-fixed ( sasha ) is true
    _____ ( _____ ) is _____
end
fun _____ ( _____ ) : _____
end
```

Define a function called `gender`, which consumes a Row of the animals table tells us the gender of that animal

|         |   |        |   |       |
|---------|---|--------|---|-------|
| _____   | : | _____  | → | _____ |
| name    |   | domain |   | range |
| # _____ |   |        |   |       |

**examples:**

```
    _____ ( _____ ) is _____
    _____ ( _____ ) is _____
end
fun _____ ( _____ ) : _____
end
```

Define a function called `is-cat`, which consumes a Row of the `animals` table and produces `true` if it's a cat.

|   |   |                          |   |                |
|---|---|--------------------------|---|----------------|
| <i>is-cat</i>   | : | ( <i>animal :: Row</i> ) | → | <i>Boolean</i> |
| name  |   | domain                   |   | range          |
| # Consumes an animal, and return true if the species is "cat" |   |                          |   |                |

**examples:**

```

    is-cat ( sasha ) is
end
fun ( ) :
end

```

Define a function called `is-young`, which consumes a Row of the `animals` table and produces `true` if it's an animal that is less than two years old.

|      |   |        |   |       |
|------|---|--------|---|-------|
|      | : |        | → |       |
| name |   | domain |   | range |
| #    |   |        |   |       |

**examples:**

```

    ( ) is
end
fun ( ) :
end

```

---

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

|                      |   |                              |   |                    |
|----------------------|---|------------------------------|---|--------------------|
| <code>nametag</code> | : | <code>(animal :: Row)</code> | → | <code>Image</code> |
| name                 |   | domain                       |   | range              |

# *Consumes an animal, and produces an image of their name in big, red letters*

**examples:**

```
    nametag ( sasha ) 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 an

|      |   |        |   |       |
|------|---|--------|---|-------|
|      | : |        | → |       |
| name |   | domain |   | range |

# \_\_\_\_\_

**examples:**

```
    _____ ( _____ ) is _____  
    _____ ( _____ ) is _____  
end  
fun _____ ( _____ ) : _____  
end
```

# My Dataset

1. My dataset is \_\_\_\_\_

2. Four of my columns are....

(choose four columns, and for each one fill out the name, datatype, and whether it contains Qualitative or Categorical data in the table below)

|                                 |  |  |  |  |
|---------------------------------|--|--|--|--|
| <b>Name</b>                     |  |  |  |  |
| <b>Datatype</b>                 |  |  |  |  |
| <b>Quantitative/Categorical</b> |  |  |  |  |

3. Three questions I have about my dataset:

1.

---

---

2.

---

---

3.

---

---



# Unit 3

- **Methods** are special functions that are attached to pieces of data. We use them to manipulate Tables.
- They are different from functions in several ways:
  1. Their names can't be used alone: they can only be used as part of data, separated by a dot. (For example, `animals.order-by`)
  2. Their contracts are different: they include the type of the data as part of their names. (eg, `<table>.order-by :: (column :: String) → Table`)
  3. They have a "secret" argument, which is the data they are attached to
- We will use three **Table Methods** to manipulate our datasets:
  1. `<Table>.order-by` – order the rows of a table based on a column
  2. `<Table>.filter` – create a **subset** of the data, with only certain rows
  3. `<Table>.build-column` – use the columns of a table to compute a new one
- We use **Table Plans** to help us use these methods correctly, without making mistakes





# Reviewing Functions

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.

# Plans for the `animals` Dataset

What are two ways you might want to *order* the `animals` dataset?

1) \_\_\_\_\_

2) \_\_\_\_\_

What are two subsets into which you might *filter* the `animals` dataset?

1) \_\_\_\_\_

2) \_\_\_\_\_

What are two new columns you might want to *build* from the `animals` dataset?

1) \_\_\_\_\_

2) \_\_\_\_\_

# Methods

Methods are a lot like functions, but they differ in three important ways:

- They can only be called as **part of a value**, using the **dot-accessor**. For example: `animals.row-n(2)`
- Their Contracts are different, because they contain a **Type** as part of their name. For example: `<Table>.row-n :: (index :: Number) -> Row`
- They have a “secret argument”, which is the value they are attached to. In the examples above, the `row-n` method consumes only a `Number` as part of its Domain, but it *also* consumes a `Table`.

Here is the Contract for a method, which consumes the name of a food and produces True if the person likes that food:

```
<Person>.likes :: (food :: String) -> Boolean
```

1. What Type of data is the method *attached to*? \_\_\_\_\_
2. What is the name of this method? \_\_\_\_\_
3. How many things are in its Domain? \_\_\_\_\_
4. What is the name of the argument in its Domain? \_\_\_\_\_
5. What is the Type of the argument in its Domain? \_\_\_\_\_
6. What Type of data will this method will produce? \_\_\_\_\_
7. Below are 3 expressions. Based on the contract above, circle the correct one.

```
emma.likes("pizza")
```

```
likes("pizza")
```

```
likes(emma, pizza)
```

8. On the line below, write your own expression that uses this method, replacing `emma` and `"pizza"` with your own name and a food you like.

# 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

`get-kittens-tags` : `(animals :: Table)` → `Table`

# 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.

`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     |

→ `get-kittens-tags(animals-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   |

## Define the function

Use the relevant methods (circle your helper functions!), then produce a result with the new table.

fun `get-kittens-tags` ( `pets` ) :

`t = pets`

`.build-columns(`

`.filter(`

`.order-by(`

`t`

end

Define the table

Are there more columns?

Are there fewer rows?

Are the rows ordered?

Produce the result

# 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

`get-dogs-by-age` : `(animals :: Table)` → `Table`

*# Consume a table of animals, and produce a table containing only the dogs, sorted by age*

## Examples

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     |

| name   | species | age | fixed | legs | weight | adopt |
|--------|---------|-----|-------|------|--------|-------|
| 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(` \_\_\_\_\_ )

*Are there more columns?*

\_\_\_\_\_ `.filter(` \_\_\_\_\_ )

*Are there fewer rows?*

\_\_\_\_\_ `.order-by(` \_\_\_\_\_ )

*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.

animals-table



old-dogs-diet(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     |

| 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     |

## Define the function

Use the relevant methods (circle your helper functions!), then produce a result with the new table.

**fun** \_\_\_\_\_ ( \_\_\_\_\_ ) :

*t =* \_\_\_\_\_

\_\_\_\_\_ *.build-columns(* \_\_\_\_\_ *)*

*Are there more columns?*

\_\_\_\_\_ *.filter(* \_\_\_\_\_ *)*

*Are there fewer rows?*

\_\_\_\_\_ *.order-by(* \_\_\_\_\_ *)*

*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.

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     |

| 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 |

## Define the function

Use the relevant methods (circle your helper functions!), then produce a result with the new table.

fun \_\_\_\_\_ ( \_\_\_\_\_ ) :

t =

.build-columns( \_\_\_\_\_ )

*Are there more columns?*

.filter( \_\_\_\_\_ )

*Are there fewer rows?*

.order-by( \_\_\_\_\_ )

*Are the rows ordered?*

\_\_\_\_\_

Produce the result

end

# My Dataset

**What are two ways you might want to *order* this dataset?**

1) \_\_\_\_\_

2) \_\_\_\_\_

**What are two subsets into which you might *filter* this dataset?**

1) \_\_\_\_\_

2) \_\_\_\_\_

**What are two new columns you might want to *build* from this dataset?**

1) \_\_\_\_\_

2) \_\_\_\_\_



# Unit 4

- **Bar charts** show the *absolute* quantity of each row in a dataset. The larger the quantity, the longer the bar. Bar charts provide a visual representation of values in a dataset.
- **Pie charts** show the *relative* quantity of each row in a dataset. The greater the percentage, the larger the pie slice. Pie charts provide a visual representation of proportions in a dataset.
- **Choosing a Sample Table** is important when coming up with small examples for Table Plans. A good sample table has:
  1. At least all the relevant columns
  2. Enough rows to accurately represent the dataset
  3. Rows that are randomly-ordered



# Statements about Columns

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/are the heaviest? \_\_\_\_\_

2. Which animal(s) is/are 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?

---

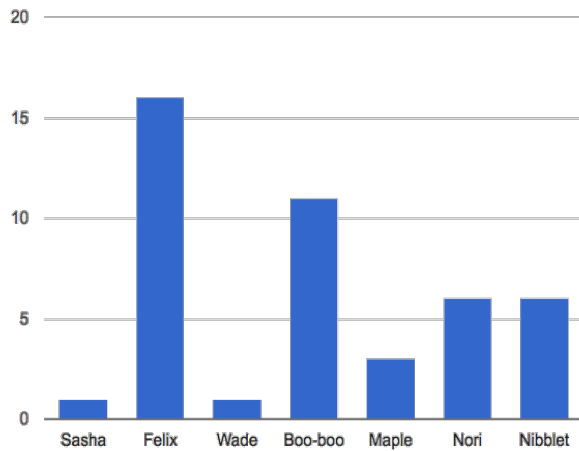
---

---

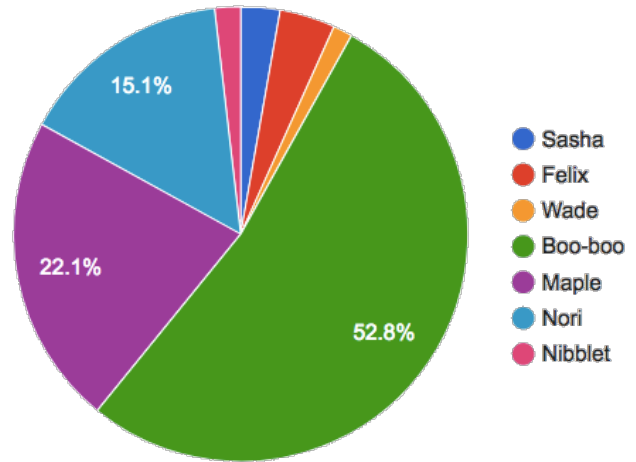
---

# Visualizing Quantity

In the table below, there are two observations drawn from the following charts. Add two more.



Animals Ages (yrs)



Animals Weights (lbs)

| Based on a ____ chart of ____                     | I notice that ____                                       |
|---|--|
| Based on a <b>bar chart</b> of 7 animals' ages    | Felix is by far the oldest                               |
| Based on a <b>pie chart</b> of 7 animals' weights | Boo-boo weighs more than the other six animals combined! |
| Based on a <b>bar chart</b> of 7 animals' ages    |  |
| Based on a <b>pie chart</b> of 7 animals' weights |  |

# 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** \_\_\_\_\_ ( \_\_\_\_\_ ) :

*t* = \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**end**

Define the table

*Are there more columns?*

*Are there fewer rows?*

*Are the rows ordered?*

Produce the result

# Bad Sample Tables!

For each word problem, a Sample Table must have (1) all the columns that matter, (2) a representative sample of the rows, and be in (3) random order. For each problem below, check the boxes to determine if the Sample 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     |
| Sunflower | cat     | 5   | TRUE  | 4    | 8.1    | 10    |

- ☐ Relevant columns
- ☐ Representative sample of rows
- ☐ Random order

## 2. The shelter wants a pie chart showing all the dogs' weight

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

- ☐ Relevant columns
- ☐ Representative sample of rows
- ☐ Random order

## 3. 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

## 4. 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.   |  |  |  |  |  |  |  |
| _____  | → _____  |  |  |  |  |  |  |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr style="background-color: #cccccc;"> <td style="height: 20px;"></td> </tr> <tr><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td></tr> </table> |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| Define the function  |  |  |  |  |  |  |  |
| Use the relevant methods (circle your helper functions!), then produce a result with the new table.  |  |  |  |  |  |  |  |
| <pre> <b>fun</b> _____ ( _____ ) :   <i>t</i> = _____   _____   _____   _____   _____   _____ <b>end</b>           </pre>  | <p style="text-align: right;"><u>Define the table</u></p> <p style="text-align: right;"><i>Are there more columns?</i></p> <p style="text-align: right;"><i>Are there fewer rows?</i></p> <p style="text-align: right;"><i>Are the rows ordered?</i></p> <p style="text-align: right;"><u>Produce the result</u></p> |  |  |  |  |  |  |

# 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** \_\_\_\_\_ ( \_\_\_\_\_ ) :

*t* =

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Define the table

*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** \_\_\_\_\_ ( \_\_\_\_\_ ) :

*t* = \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Define the table

*Are there more columns?*

*Are there fewer rows?*

*Are the rows ordered?*

Produce the result

**end**

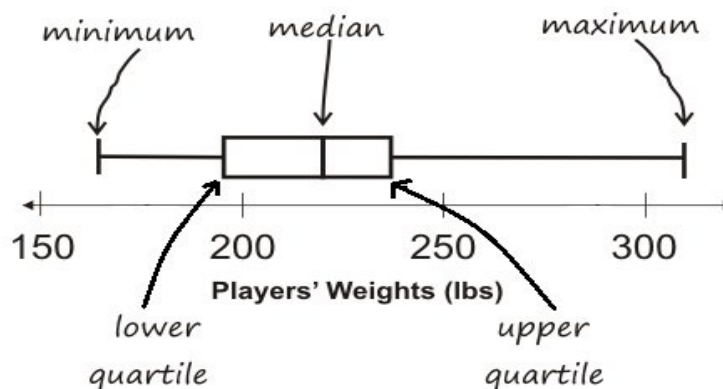
# Visualizing My Dataset

What quantity charts did you make, and what do you notice? Fill in the table below.

| Based on a ____ chart of ____       | I notice that ____                  |
|-------------------------------------|-------------------------------------|
| <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> | <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> |
| <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> | <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> |
| <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> | <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> |
| <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> | <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> |

# Unit 5

- There are three ways to measure the “center” of a dataset, to talk about a whole column of data using just one number:
  1. The **mean** of a dataset is the average of all the numbers
  2. The **median** of a dataset is a value that is smaller than half the dataset, and larger than the other half
  3. The **modes** of a dataset are the numbers that appear the most often.
- Data Scientists can also measure the “variation” of a dataset using a **five number summary**:
  1. The **minimum** – the smallest value in the dataset
  2. The **first, or “lower” quartile (Q1)** – the median value that separates the first quarter of the values in the dataset from the second quarter
  3. The **second quartile (Q2)** – the median value which separates the entire dataset into “top” and “bottom” halves.
  4. The **third, or “upper” quartile (Q3)** – the median value that separates the third quarter of the values in the dataset from the fourth quarter
  5. The **maximum** – the largest value in the dataset
- The **five number summary** can be used to draw a **box-and-whisker plot**.





# Summarizing Columns in Animals

The column I choose to measure is weeks

## Measures of Center

The three measures for this column are:

| Mean (Average) | Median | Mode(s) |
|----------------|--------|---------|
|                |        |         |

Based on the differences between mean and median, I conclude :

---

---

---

---

## Measures of Variation

My five-number summary is:

| Minimum | Q1 | Q2 (Median) | Q3 | Maximum |
|---------|----|-------------|----|---------|
|         |    |             |    |         |

A box plot can be drawn from this summary on the number line below:



From this summary and box-plot, I conclude:

---

---

---

---

# Table Plan

The shelter wants a summary of the variation in ages among the dogs. Write a function called `variation-dog-age` that will take in a table of animals produce a box-plot that shows this variation.

## Contract and Purpose

\_\_\_\_\_ : \_\_\_\_\_ → \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

## Examples

Make a Start Table and a result based on that table.

animals-table

→ variation-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 \_\_\_\_\_ ( \_\_\_\_\_ ) :

*t =*

*.build-columns(* \_\_\_\_\_ *)*

*Are there more columns?*

*.filter(* \_\_\_\_\_ *)*

*Are there fewer rows?*

*.order-by(* \_\_\_\_\_ *)*

*Are the rows ordered?*

\_\_\_\_\_

*Produce the result*

end

# Interpreting Variation

Consider the following list dataset, representing the annual income of ten people:

\$65k, \$12k, \$14k, \$280k, \$15k, \$22k, \$45k, \$34k, \$45k, \$175k

1. In the space below, rewrite this dataset in **sorted order**.

2. In the table below, compute the **measures of center** for this dataset.

| Mean (Average) | Median | Mode(s) |
|----------------|--------|---------|
|                |        |         |

3. In the table below, compute the **five number summary** of this dataset.

| Minimum | Q1 | Q2 (Median) | Q3 | Maximum |
|---------|----|-------------|----|---------|
|         |    |             |    |         |

4. On the number line below, draw a **box plot** for this dataset.



5. The following statements are *correct*...but misleading. Write down the reason why.

| Statement  | Why it's misleading |
|--|---------------------|
| "The average person makes more than \$70k dollars!"  |                     |
| "The most common salary is \$45k/yr!"  |                     |
| "This group is really diverse, with people making as little as 12k and as much as \$280k!" |                     |

# Summarizing a Column in My Dataset

The column I choose to measure is \_\_\_\_\_

## Measures of Center

The three measures for this column are:

| Mean (Average) | Median | Mode(s) |
|----------------|--------|---------|
|                |        |         |

Based on the differences between mean and median, I conclude :

---

---

---

---

## Measures of Variation

My five-number summary is:

| Minimum | Q1 | Q2 (Median) | Q3 | Maximum |
|---------|----|-------------|----|---------|
|         |    |             |    |         |

A box plot can be drawn from this summary on the number line below:



From this summary and box-plot, I conclude:

---

---

---

---



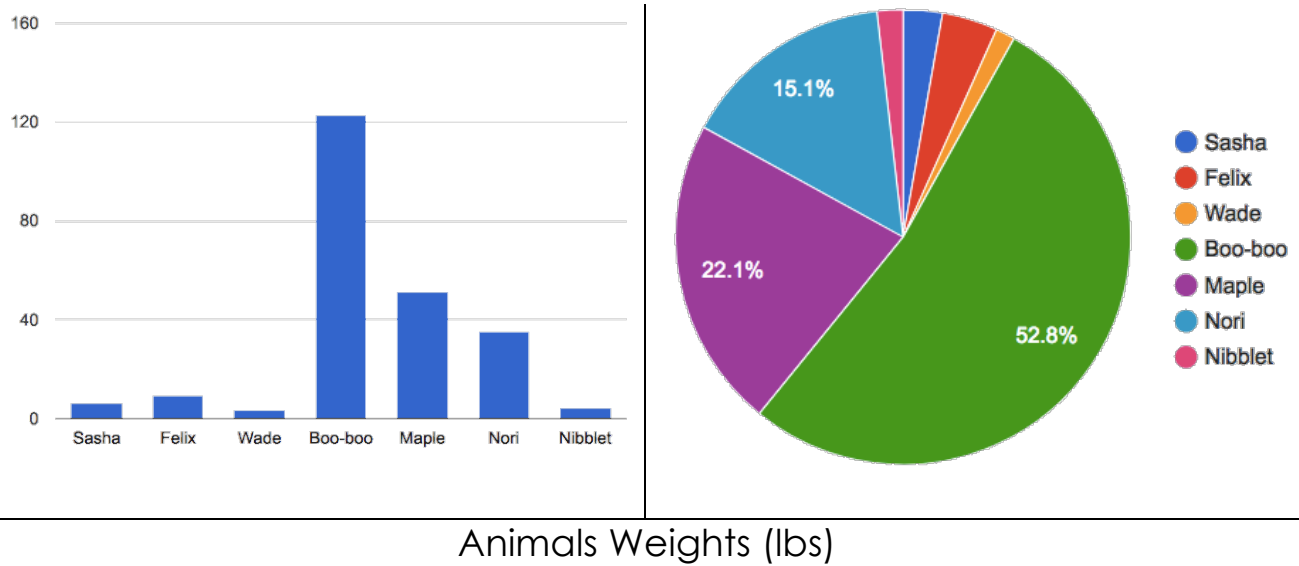
# Unit 6

- **Frequency Bar charts** show the number of rows belonging to a given category. The more rows in each category, the longer the bar. Frequency bar charts provide a visual representation of the frequency of values in a **categorical** column. Since categorical data cannot be ordered, there is no strict ordering of bars in a frequency bar chart.
- **Histograms** show the number of rows that fall within certain ranges, or “bins” of a dataset. The more rows that fall within a particular “bin”, the longer the bar. Histograms provide a visual representation of the frequency of values in a **quantitative** column. Quantitative data can be ordered, so the bars of a histogram are always sorted.
- When dealing with histograms, it's important to select a good **bin size**. If the bins are too small or too large, it is difficult to see the distribution in the dataset.



# Visualizing Quantity (Review)

Use the charts below to help you answer the questions.



1. Which animal is the heaviest? \_\_\_\_\_

2. Which animal is the lightest? \_\_\_\_\_

3. How much of the *total weight* comes from Maple? \_\_\_\_\_

4. How much of the *total weight* comes from Nori? \_\_\_\_\_

5. Which chart did you use for questions 1 and 2? \_\_\_\_\_

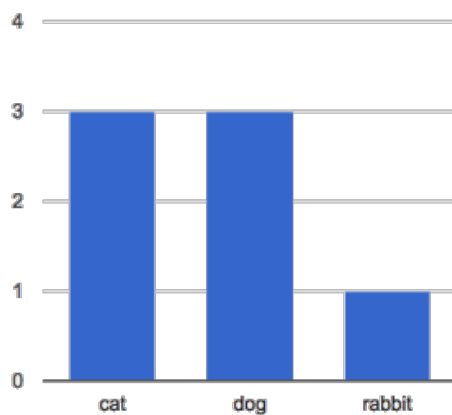
6. Which chart did you use for questions 3 and 4? \_\_\_\_\_

7. Why are some questions easier to answer with one kind of chart or another? \_\_\_\_\_

# 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   |

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




---



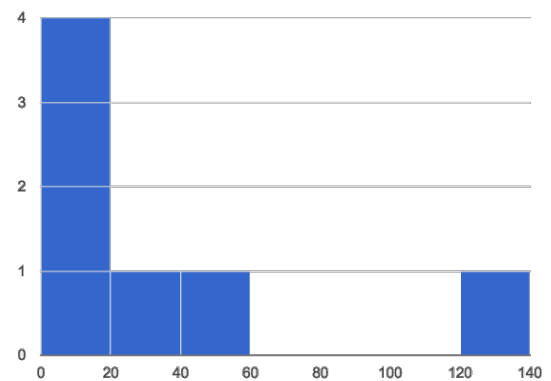
---



---



---




---



---



---



---

# Table Plan

Define a function `freq-bar-gender`, which takes in a Table of animals and creates a frequency bar chart showing how many animals are male v. female.

| Contract and Purpose   |   |  |  |  |  |  |  |
|--|---|--|--|--|--|--|--|
| _____ :  | → _____   |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |
| Examples   |   |  |  |  |  |  |  |
| Make a Start Table and a result based on that table.   |   |  |  |  |  |  |  |
| _____  | → _____   |  |  |  |  |  |  |
| <table border="1" style="width: 100%; border-collapse: collapse;"> <tr style="background-color: #808080;"> <td style="height: 20px;"></td> </tr> <tr><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td></tr> <tr><td style="height: 20px;"></td></tr> </table> |   |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |
|  |   |  |  |  |  |  |  |
| Define the function  |   |  |  |  |  |  |  |
| Use the relevant methods (circle your helper functions!), then produce a result with the new table.  |   |  |  |  |  |  |  |
| <pre> fun _____ ( _____ ) :   <i>t</i> = _____   _____   _____   _____   _____ end </pre>  | <p><u>Define the table</u></p> <p><i>Are there more columns?</i></p> <p><i>Are there fewer rows?</i></p> <p><i>Are the rows ordered?</i></p> <p><u>Produce the result</u></p> |  |  |  |  |  |  |

# Table Plan

Define a function `histogram-adoption`, which takes in a Table of animals and creates a histogram showing how long it took for animals to get adopted

| Contract and Purpose   |                           |  |  |  |  |  |  |
|--|---------------------------|--|--|--|--|--|--|
| _____  | : _____ → _____           |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |
| Examples   |                           |  |  |  |  |  |  |
| Make a Start Table and a result based on that table.   |                           |  |  |  |  |  |  |
| _____  | → _____                   |  |  |  |  |  |  |
| <table border="1"><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr><tr><td></td></tr></table> |                           |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |
|  |                           |  |  |  |  |  |  |
| Define the function  |                           |  |  |  |  |  |  |
| Use the relevant methods (circle your helper functions!), then produce a result with the new table.                                    |                           |  |  |  |  |  |  |
| <code>fun</code> _____ ( _____ ) :   | <u>Define the table</u>   |  |  |  |  |  |  |
| <code>t</code> = _____   | Are there more columns?   |  |  |  |  |  |  |
| _____  | Are there fewer rows?     |  |  |  |  |  |  |
| _____  | Are the rows ordered?     |  |  |  |  |  |  |
| _____  | <u>Produce the result</u> |  |  |  |  |  |  |
| <code>end</code>   |                           |  |  |  |  |  |  |

# Visualizing My Dataset

What frequency charts did you make, and what do you notice? Fill in the table below.

| Based on a ____ chart of ____             | I notice that ____                        |
|---|---|
| <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> | <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> |
| <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> | <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> |
| <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> | <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> |
| <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> | <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> <hr/> |

# Matching Charts to Questions

For each of the questions below, draw a line to the chart that will best answer it.

|   |                            |
|---|----------------------------|
| 1. Are there more of the animals at the shelter fixed or unfixed?           | <b>Pie Chart</b>           |
| 2. How many weeks did each cat wait to be adopted?                          | <b>Bar Chart</b>           |
| 3. How many male v. female dogs are there?                                  | <b>Frequency Bar Chart</b> |
| 4. How many animals have 4 legs? 8? 3?                                      | <b>Histogram</b>           |
| 5. What percent of the total weight at the shelter is made up by Boo-boo?   |                            |
| 6. What is the distribution of weights across all the animals older than 3? |                            |
| 7. How many animals are there of each species?                              |                            |
| 8. Who waited the longest to be adopted?                                    |                            |



# Unit 7

- **Scatter Plots** show the relationship between two quantitative columns. Each row in the dataset is represented by a point, with one column providing the x-value and the other providing the y-value. The resulting “point cloud” makes it possible to look for a relationship between those two columns.
- If the points in a scatter plot appear to follow a pattern, it is possible that a relationship – or **correlation** – exists between those two columns.
- If there is a pattern to the points in a scatter plot, points that are far away from the pattern are called **outliers**.
- We can express this correlation by drawing line through the data cloud, so that the distance between the line and each of the points is as small as possible. This line is called the **line of best fit** – or **predictor function** - and allows us to make predictions based on the dataset.



# (Dis)Proving a Claim

***“Younger animals are cuter, so they get adopted faster.”***

*Do you agree? If so, why?*

*I hypothesize...*

---

---

---

---

---

---

---

---

---

---

*What would you look for in the dataset to see if you are right?*

---

---

---

---

---

---

---

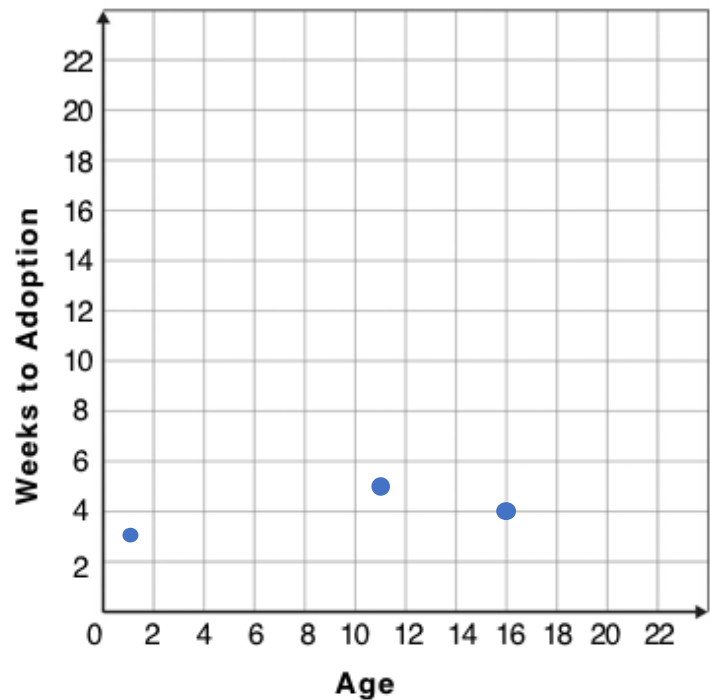
---

---

---

# Creating a Scatter Plot

| name      | species  | age | weeks |
|-----------|----------|-----|-------|
| "Sasha"   | "cat"    | 1   | 3     |
| "Boo-boo" | "dog"    | 11  | 5     |
| "Felix"   | "cat"    | 16  | 4     |
| "Buddy"   | "lizard" | 2   | 24    |
| "Nori"    | "dog"    | 6   | 9     |
| "Wade"    | "cat"    | 1   | 2     |
| "Nibblet" | "rabbit" | 6   | 12    |
| "Maple"   | "dog"    | 3   | 2     |



1. **For each row in the Sample Table on the left, add a point to the scatter plot on the right.** The first 3 rows have been completed for you. Use the values from the `age` column for the x-axis, and values from the `weeks` column for the y-axis.
2. Do you see a pattern? Do the points seem to shift up or down as age increases?  
**Draw a line on the scatter plot to show this pattern.**
3. Does the line slope upwards or downwards? \_\_\_\_\_
4. Are the points mostly close to the line? \_\_\_\_\_

# Table Plan

Define a function `dogs-age-weeks`, which takes in a Table of animals and creates a scatter plot of all the dogs, tracking their `age` on the x-axis and the number of `weeks` it took for them to be adopted on the y-axis.

## 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** \_\_\_\_\_ ( \_\_\_\_\_ ) :

*t =* \_\_\_\_\_

|       |
|-------|
| _____ |
| _____ |
| _____ |
| _____ |
| _____ |

Define the table

*Are there more columns?*

*Are there fewer rows?*

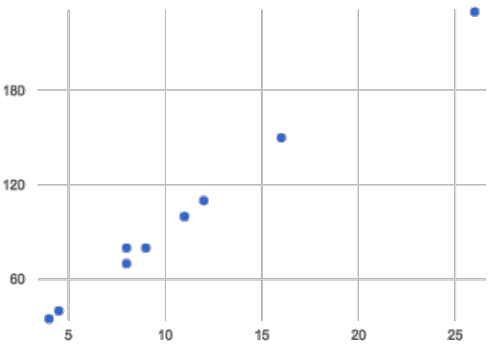
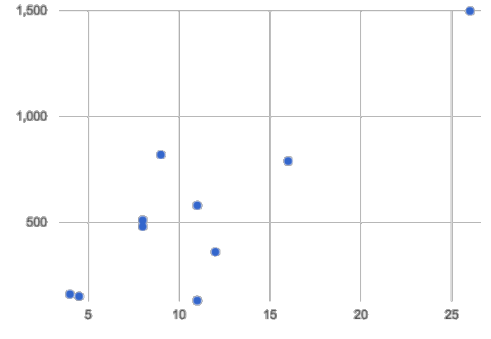
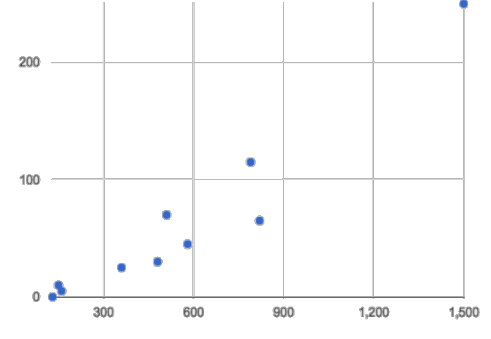
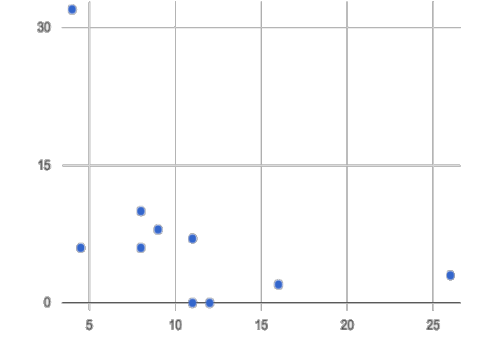
*Are the rows ordered?*

Produce the result

**end**

# Drawing Predictors

For each of the scatter plots below, draw a **predictor line** that fits best.

|   |  |   |
|---|--|---|
| A |  <p>fat (g) v. calories-from-fat in common menu items</p>     | <p><b>Direction:</b> Positive   Negative   None</p> <p><b>Strength:</b> Strong   Weak</p> |
| B |  <p>fat (g) v. sodium (g) in common menu items</p>           | <p><b>Direction:</b> Positive   Negative   None</p> <p><b>Strength:</b> Strong   Weak</p> |
| C |  <p>sodium (g) v. cholesterol (mg) in common menu items</p> | <p><b>Direction:</b> Positive   Negative   None</p> <p><b>Strength:</b> Strong   Weak</p> |
| D |  <p>fat (g) v. sugar (g) in common menu items</p>           | <p><b>Direction:</b> Positive   Negative   None</p> <p><b>Strength:</b> Strong   Weak</p> |

# Correlations in My Dataset

1) There may be a correlation between \_\_\_\_\_ and  
column  
\_\_\_\_\_. I think it is a \_\_\_\_\_,  
column strong / weak positive / negative  
correlation, because \_\_\_\_\_  
\_\_\_\_\_. It would be stronger if I looked  
at \_\_\_\_\_.  
a subset or extension of my data

---

1) There may be a correlation between \_\_\_\_\_ and  
column  
\_\_\_\_\_. I think it is a \_\_\_\_\_,  
column strong / weak positive / negative  
correlation, because \_\_\_\_\_  
\_\_\_\_\_. It would be stronger if I looked  
at \_\_\_\_\_.  
a subset or extension of my data

---

1) There may be a correlation between \_\_\_\_\_ and  
column  
\_\_\_\_\_. I think it is a \_\_\_\_\_,  
column strong / weak positive / negative  
correlation, because \_\_\_\_\_  
\_\_\_\_\_. It would be stronger if I looked  
at \_\_\_\_\_.  
a subset or extension of my data





# Unit 8

- Given a **predictor function** and a scatter plot, we can compute the error by adding the squares of all the distances between the function and each point in the plot. The error is called the  **$r^2$  statistic**, which tells us *how much of the variation in the y-axis can be explained by the x-axis*.
- A **strong correlation** will have a large  $r^2$ . A **weak correlation** will have a small  $r^2$ .
- A **positive correlation** means the slope of the line of best fit is positive. A **negative correlation** means the slope is negative.
- **Linear Regression** is a way of computing the **line of best fit**, by taking a scatter plot and deriving the slope and y-intercept for a line that has the smallest possible  $r^2$ .
- **Correlation is not causation!** Correlation only suggests that two measures are *related*, but does not tell us if one *causes* the other. For example, hot days are *correlated* with people running their air conditioners, air conditioners do not *cause* hot days!

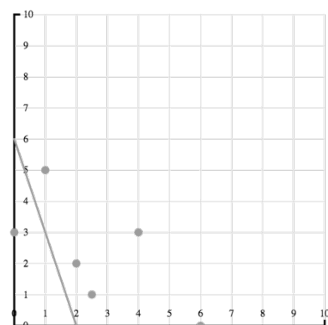
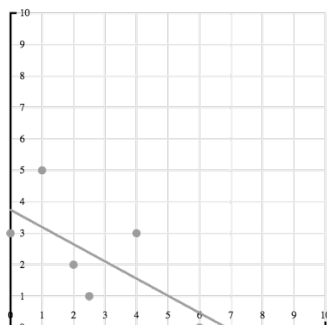


# Grading Predictors

Below are the scatter plots for data sets A-D, with two different lines predictor lines drawn on top. For plots A-D:

1. Circle the plot with the line that fits better
2. Give the plot you circled a grade between 0 (no correlation) and 1 (perfect correlation)

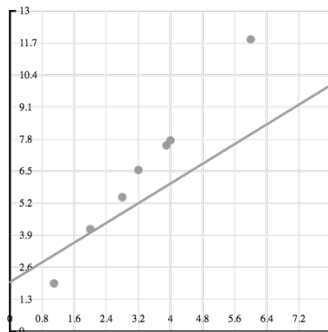
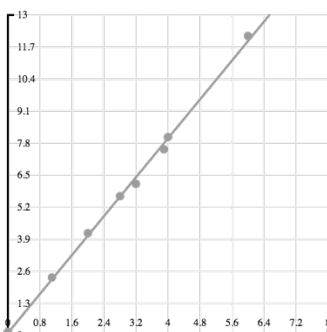
**A**



Strength of  
Correlation:

\_\_\_\_\_

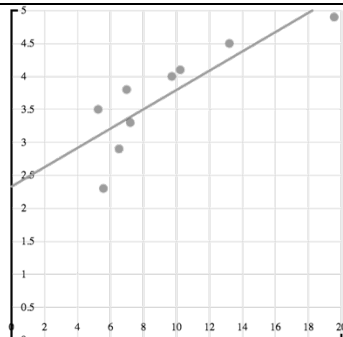
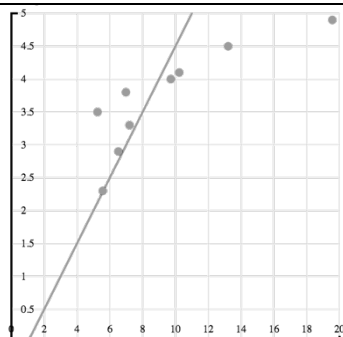
**B**



Strength of  
Correlation:

\_\_\_\_\_

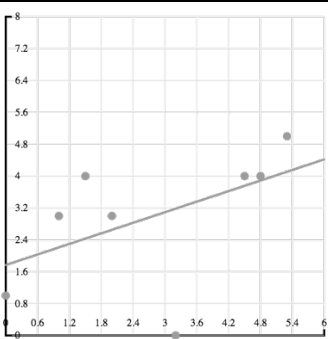
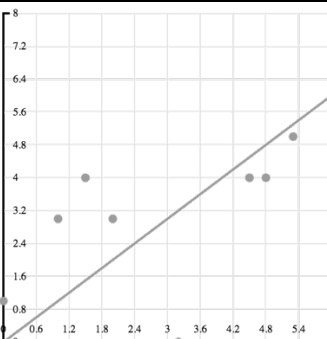
**C**



Strength of  
Correlation:

\_\_\_\_\_

**D**



Strength of  
Correlation:

\_\_\_\_\_

# Findings in the animals Dataset

I performed a linear regression on \_\_\_\_\_, and  
dataset or subset  
found \_\_\_\_\_ correlation between  
a strong/weak ( $r^2=...$ ), positive/negative  
\_\_\_\_\_ and \_\_\_\_\_. From this, I  
[x-axis] [y-axis]  
conclude that \_\_\_\_\_  
25% of the variability in adoption time is explained  
 $r^2$  % of the variation in [y-axis] is explained by [x-axis]  
by the age of the dog

---

I performed a linear regression on \_\_\_\_\_, and  
dataset or subset  
found \_\_\_\_\_ correlation between  
a strong/weak ( $r^2=...$ ), positive/negative  
\_\_\_\_\_ and \_\_\_\_\_. From this, I  
[x-axis] [y-axis]  
conclude that \_\_\_\_\_  
 $r^2$  % of the variation in [y-axis] is explained by [x-axis]  
\_\_\_\_\_

---

I performed a linear regression on \_\_\_\_\_, and  
dataset or subset  
found \_\_\_\_\_ correlation between  
a strong/weak ( $r^2=...$ ), positive/negative  
\_\_\_\_\_ and \_\_\_\_\_. From this, I  
[x-axis] [y-axis]  
conclude that \_\_\_\_\_  
 $r^2$  % of the variation in [y-axis] is explained by [x-axis]  
\_\_\_\_\_

---

# Correlations in My Dataset

I performed a linear regression on \_\_\_\_\_, and  
dataset or subset  
found \_\_\_\_\_ correlation between  
a strong/weak ( $r^2=...$ ), positive/negative  
\_\_\_\_\_ and \_\_\_\_\_. From this, I  
[x-axis] [y-axis]  
conclude that \_\_\_\_\_  
 $r^2$  % of the variation in [y-axis] is explained by [x-axis]  
\_\_\_\_\_.

---

I performed a linear regression on \_\_\_\_\_, and  
dataset or subset  
found \_\_\_\_\_ correlation between  
a strong/weak ( $r^2=...$ ), positive/negative  
\_\_\_\_\_ and \_\_\_\_\_. From this, I  
[x-axis] [y-axis]  
conclude that \_\_\_\_\_  
 $r^2$  % of the variation in [y-axis] is explained by [x-axis]  
\_\_\_\_\_.

---

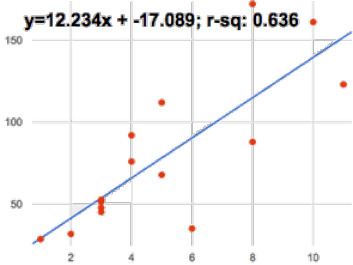
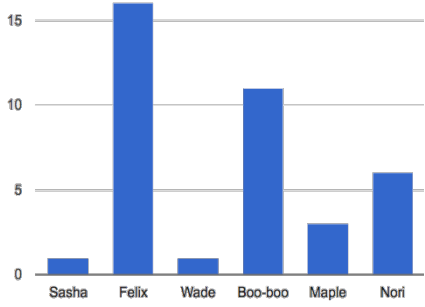
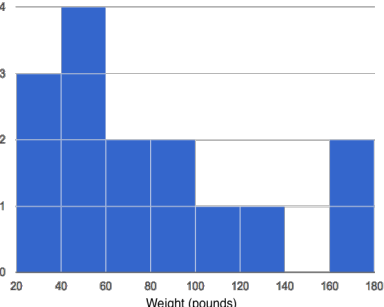
I performed a linear regression on \_\_\_\_\_, and  
dataset or subset  
found \_\_\_\_\_ correlation between  
a strong/weak ( $r^2=...$ ), positive/negative  
\_\_\_\_\_ and \_\_\_\_\_. From this, I  
[x-axis] [y-axis]  
conclude that \_\_\_\_\_  
 $r^2$  % of the variation in [y-axis] is explained by [x-axis]  
\_\_\_\_\_.

---

## Unit 9

# Fake News!

**Every claim below is *wrong*!** Your job is to figure out why, by looking at the data.

|   | Data   | Claim  | Why it's wrong |
|---|--|--|----------------|
| 1 | The average player on a basketball team is 6'1".   | "Most of the players on the team are taller than 6'."  |                |
| 2 | After performing linear regression on census data, a positive correlation ( $r^2=0.18$ ) was found between people's height and salary.                                   | "Taller people get paid more."   |                |
| 3 |    | "According to the predictor function indicated here, the value on the x-axis is will predict the value on the y-axis 63.6% of the time." |                |
| 4 | <br>Bar Chart of Pet Ages   | "According to this bar chart, Felix makes up a little more than 15% of the total ages of all the animals in the dataset."                |                |
| 5 | <br>Weight (pounds)   | "According to this histogram, most animals weigh between 40 and 60 pounds."  |                |
| 6 | After performing linear regression, a negative correlation ( $r^2=0.91$ ) was found between the number of hairs on a person's head and their likelihood of owning a wig. | "Owning wigs causes people to go bald."  |                |

# **Blank Recipes, Table Plans, and References**



# Design Recipes

---

|       |       |        |   |       |
|-------|-------|--------|---|-------|
| _____ | :     | _____  | → | _____ |
| name  |       | domain |   | range |
| #     | _____ |        |   |       |

**examples:**

```
    _____ ( _____ ) is _____  
    _____ ( _____ ) is _____  
end  
fun _____ ( _____ ) : _____  
end
```

---

|       |       |        |   |       |
|-------|-------|--------|---|-------|
| _____ | :     | _____  | → | _____ |
| name  |       | domain |   | range |
| #     | _____ |        |   |       |

**examples:**

```
    _____ ( _____ ) is _____  
    _____ ( _____ ) is _____  
end  
fun _____ ( _____ ) : _____  
end
```

# Design Recipes

---

|       |       |        |   |       |
|-------|-------|--------|---|-------|
| _____ | :     | _____  | → | _____ |
| name  |       | domain |   | range |
| #     | _____ |        |   |       |

**examples:**

```
    _____ ( _____ ) is _____  
end _____ ( _____ ) is _____  
fun _____ ( _____ ) : _____  
end
```

---

|       |       |        |   |       |
|-------|-------|--------|---|-------|
| _____ | :     | _____  | → | _____ |
| name  |       | domain |   | range |
| #     | _____ |        |   |       |

**examples:**

```
    _____ ( _____ ) is _____  
end _____ ( _____ ) is _____  
fun _____ ( _____ ) : _____  
end
```

# Design Recipes

---

|       |       |        |   |       |
|-------|-------|--------|---|-------|
| _____ | :     | _____  | → | _____ |
| name  |       | domain |   | range |
| #     | _____ |        |   |       |

**examples:**

```
    _____ ( _____ ) is _____  
    _____ ( _____ ) is _____  
end  
fun _____ ( _____ ) : _____  
end
```

---

|       |       |        |   |       |
|-------|-------|--------|---|-------|
| _____ | :     | _____  | → | _____ |
| name  |       | domain |   | range |
| #     | _____ |        |   |       |

**examples:**

```
    _____ ( _____ ) is _____  
    _____ ( _____ ) is _____  
end  
fun _____ ( _____ ) : _____  
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** \_\_\_\_\_ ( \_\_\_\_\_ ) :

*t* = \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Define the table

*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** \_\_\_\_\_ ( \_\_\_\_\_ ) :

*t* =

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Define the table

*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** \_\_\_\_\_ ( \_\_\_\_\_ ) :

*t* = \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

Define the table

*Are there more columns?*

*Are there fewer rows?*

*Are the rows ordered?*

Produce the result

**end**

# 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  |
| string-contains | :: (text :: String, search-for :: String)                    | → Boolean |
| 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                     |
|---|---|---------------|---------------------------|
| <code>&lt;Table&gt;.row-n</code>        | <code>:: (n :: Number)</code>   | $\rightarrow$ | <i>Row</i>                |
| <code>&lt;Table&gt;.filter</code>       | <code>:: (test :: (Row <math>\rightarrow</math> Boolean) )</code>                 | $\rightarrow$ | <i>Table</i>              |
| <code>&lt;Table&gt;.build-column</code> | <code>:: (col :: String, builder :: (Row <math>\rightarrow</math> Value) )</code> | $\rightarrow$ | <i>Table</i>              |
| <code>mean</code>                       | <code>:: (t :: Table, col :: String)</code>                                       | $\rightarrow$ | <i>Number</i>             |
| <code>median</code>                     | <code>:: (t :: Table, col :: String)</code>                                       | $\rightarrow$ | <i>Number</i>             |
| <code>modes</code>                      | <code>:: (t :: Table, col :: String)</code>                                       | $\rightarrow$ | <i>List&lt;Number&gt;</i> |
| <code>bar-chart</code>                  | <code>:: (t :: Table, labels :: String, values :: String)</code>                  | $\rightarrow$ | <i>Image</i>              |
| <code>pie-chart</code>                  | <code>:: (t :: Table, labels :: String, values :: String)</code>                  | $\rightarrow$ | <i>Image</i>              |
| <code>box-plot</code>                   | <code>:: (t :: Table, col :: String)</code>                                       | $\rightarrow$ | <i>Image</i>              |
| <code>freq-bar-chart</code>             | <code>:: (t :: Table, values :: String)</code>                                    | $\rightarrow$ | <i>Image</i>              |
| <code>histogram</code>                  | <code>:: (t :: Table, values :: String, bin-width :: Number)</code>               | $\rightarrow$ | <i>Image</i>              |
| <code>scatter-plot</code>               | <code>:: (t :: Table, xs :: String, ys :: String)</code>                          | $\rightarrow$ | <i>Image</i>              |
| <code>labeled-scatter-plot</code>       | <code>:: (t :: Table, labels :: String, xs :: String, ys :: String)</code>        | $\rightarrow$ | <i>Image</i>              |
| <code>lr-plot</code>                    | <code>:: (t :: Table, xs :: String, ys :: String)</code>                          | $\rightarrow$ | <i>Image</i>              |
| <code>labeled-lr-plot</code>            | <code>:: (t :: Table, labels :: String, xs :: String, ys :: String)</code>        | $\rightarrow$ | <i>Image</i>              |
|   |   |               |                           |