Name: ____

Student Workbook

Spring, 2022 - Pyret Edition

Workbook v1.5

Brought to you by the Bootstrap team:

- Emmanuel Schanzer
- Kathi Fisler
- Shriram Krishnamurthi
- Dorai Sitaram
- Joe Politz
- Ben Lerner
- Nancy Pfenning
- Flannery Denny

Visual Designer: Colleen Murphy

Introduction to Computational Data Science

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 about 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 numeric and measures *an amount*, such as a person's height, a score on a test, distance, etc. A list of quantitative data can be ordered from smallest to largest.
- **Categorical data** is data that specifies *qualities*, such as sex, eye color, country of origin, etc. Categorical data is not subject to the laws of arithmetic — for example, we cannot take the "average" of a list of colors.

The Animals Dataset

name	species	sex	age	fixed	legs	pounds	weeks
Sasha	cat	female	1	false	4	6.5	3
Snuffles	rabbit	female	3	true	4	3.5	8
Mittens	cat	female	2	true	4	7.4	1
Sunflower	cat	female	5	true	4	8.1	6
Felix	cat	male	16	true	4	9.2	5
Sheba	cat	female	7	true	4	8.4	6
Billie	snail	hermaphrodite	0.5	false	0	0.1	3
Snowcone	cat	female	2	true	4	6.5	5
Wade	cat	male	1	false	4	3.2	1
Hercules	cat	male	3	false	4	13.4	2
Toggle	dog	female	3	true	4	48	1
Boo-boo	dog	male	11	true	4	123	24
Fritz	dog	male	4	true	4	92	3
Midnight	dog	female	5	false	4	112	4
Rex	dog	male	1	false	4	28.9	9
Gir	dog	male	8	false	4	88	5
Max	dog	male	3	false	4	52.8	8
Nori	dog	female	3	true	4	35.3	1
Mr. Peanutbutter	dog	male	10	false	4	161	6
Lucky	dog	male	3	true	3	45.4	9
Kujo	dog	male	8	false	4	172	30
Buddy	lizard	male	2	false	4	0.3	3
Gila	lizard	female	3	true	4	1.2	4
Bo	dog	male	8	true	4	76.1	10
Nibblet	rabbit	male	6	false	4	4.3	2
Snuggles	tarantula	female	2	false	8	0.1	1
Daisy	dog	female	5	true	4	68	8
Ada	dog	female	2	true	4	32	3
Miaulis	cat	male	7	false	4	8.8	4
Heathcliff	cat	male	1	true	4	2.1	2
Tinkles	cat	female	1	true	4	1.7	3
Maple	dog	female	3	true	4	51.6	4

Categorical or Quantitative?

For each piece of data below, circle whether it is **Categorical** or **Quantitative** data.

1	Hair color	categorical	quantitative
2	Age	categorical	quantitative
3	ZIP Code	categorical	quantitative
4	Year	categorical	quantitative
5	Height	categorical	quantitative
6	Sex	categorical	quantitative
7	Street Name	categorical	quantitative

For each question, circle whether it will be answered by **Categorical** or **Quantitative** data.

8	We'd like to find out the average price of cars in a lot.	categorical	quantitative
9	We'd like to find out the most popular color for cars.	categorical	quantitative
10	We'd like to find out which puppy is the youngest.	categorical	quantitative
11	We'd like to find out which cats have been fixed.	categorical	quantitative
12	We want to know which people have a ZIP code of 02907.	categorical	quantitative
13	We'd like to sort a list of phone numbers by area code.	categorical	quantitative

Questions and Column Descriptions

What questions can you ask about the animals dataset? For each question, **can it be answered by this dataset?** Make sure you have at least two questions that can be answered, and at least one that cannot.

What do you NOTICE about this dataset?	What do you WONDER about this dataset?	Answered by this dataset?
		Yes No
		Yes No
		Yes No
		Yes No
		Yes No
		Yes No
		Yes No

1. This dataset is ____Animals that came from an animal shelter____, which contains __32__ data rows.
2. Some of the columns are:
   a. ____species____, which contains ____categorical____ data. Some example values are: ____"cat", "dog", and "rabbit"____.
   b. ________, which contains ________ data. Some example values are: ________.

What's on your mind?

Introduction to Programming

The **Editor** is a software program we use to write Code. Our Editor allows us to experiment with Code on the right-hand side, in the **Interactions Area**. For Code that we want to *keep*, we can put it on the left-hand side in the **Definitions Area**. Clicking the "Run" button causes the computer to re-read everything in the Definitions Area and erase anything that was typed into the Interactions Area.

Data Types

Programming languages involve different ***data types***, such as Numbers, Strings, Booleans, and even Images.

- Numbers are values like `1`, `0.4`, `1/3`, and `-8261.003`.
  - Numbers are *usually* used for quantitative data and other values are *usually* used as categorical data.
  - In Pyret, any decimal *must* start with a 0. For example, `0.22` is valid, but `.22` is not.
- Strings are values like `"Emma"`, `"Rosanna"`, `"Jen and Ed"`, or even `"08/28/1980"`.
  - All strings *must* be surrounded in quotation marks.
- Booleans are either `true` or `false`.

All values evaluate to themselves. The program `42` will evaluate to `42`, the String `"Hello"` will evaluate to `"Hello"`, and the Boolean `false` will evaluate to `false`.

Operators

Operators (like `+`, `-`, `*`, `<`, etc.) work the same way in Pyret that they do in math.

- Operators are written between values, for example: `4 + 2`.
- In Pyret, operators must always have a space around them. `4 + 2` is valid, but `4+2` is not.
- If an expression has different operators, parentheses must be used to show order of operations. `4 + 2 + 6` and `4 + (2 * 6)` are valid, but `4 + 2 * 6` is not.

Applying Functions

Applying functions works much the way it does in math. Every function has a name, takes some inputs, and produces some output. The function name is written first, followed by a list of ***arguments*** in parentheses.

- In math this could look like $f(5)$ or $g(10, 4)$.
- In Pyret, these examples would be written as `f(5)` and `g(10, 4)`.
- Applying a function to make images would look like `star(50, "solid", "red")`.
- There are many other functions, for example `num-sqr`, `num-sqrt`, `triangle`, `square`, `string-repeat`, etc.

Functions have ***contracts***, which help explain how a function should be used. Every contract has three parts:

- The *Name* of the function - literally, what it's called.
- The *Domain* of the function - what *types of values* the function consumes, and in what order.
- The *Range* of the function - what *type of value* the function produces.

Numbers and Strings

Make sure you've loaded the code.pyret.org, (CPO) editor, clicked "Run", and are working in the ***Interactions Area***.

Numbers

1) Try typing `42` into the Interactions Area and hitting "Enter". What is the largest number the editor can handle?

2) Try typing `0.5`. Then try typing `.5`. Then try clicking on the answer. Experiment with other decimals. Explain what you understand about how decimals work in this programming language.

3) What happens if you try a fraction like `1/3`?

4) Try writing negative integers, fractions and decimals.

Strings

String values are always in quotes.

5) Is `42` the same as `"42"`? Why or why not? Write your answer below:

6) Try typing your name *(in quotes!)*.

7) Try typing a sentence like "I'm excited to learn to code!" *(in quotes!)*.

8) Try typing your name with the opening quote, but *without the closing quote.* Read the error message!

9) Now try typing your name *without any quotes.* Read the error message!

10) Explain what you understand about how strings work in this programming language.

Operators

11) Just like math, Pyret has ***operators*** like `+`, `-`, `*` and `/`. Try typing in `4 + 2`, and then `4+2` (without the spaces). What can you conclude from this?

12) Type in the following expressions, one at a time: `4 + 2 + 6`, `4 + 2 * 6`, `4 + (2 * 6)`. What do you notice?

13) Try typing in `4 + "cat"`, and then `"dog" + "cat"`. What can you conclude from this?

Booleans

Boolean-producing expressions are yes-or-no questions and will always evaluate to either `true` ("yes") or `false` ("no"). What will each of the expressions below evaluate to? *Write down your prediction in the blanks provided and then type the code into the interactions area to see what it returns.*

	Prediction:	Computer Returns:		Prediction:	Computer Returns:
1) `3 <= 4`	______	______	2) `"a" > "b"`	______	______
3) `3 == 2`	______	______	4) `"a" < "b"`	______	______
5) `2 < 4`	______	______	6) `"a" == "b"`	______	______
7) `5 >= 5`	______	______	8) `"a" <> "a"`	______	______
9) `4 >= 6`	______	______	10) `"a" >= "a"`	______	______
11) `3 <> 3`	______	______	12) `"a" <> "b"`	______	______

13) In your own words, describe what `<` does.

____

14) In your own words, describe what `>=` does.

____

15) In your own words, describe what `<>` does.

____

	Prediction:	Computer Returns:
16) `string-contains("catnap", "cat")`	______	______
17) `string-contains("cat", "catnap")`	______	______

18) How many **Numbers** are there in the entire universe? ______________

19) How many **Strings** are there in the entire universe? ______________

20) How many **Images** are there in the entire universe? ______________

21) How many **Booleans** are there in the entire universe? ______________

Applying Functions

Type this line of code into the interactions area and hit "Enter":

```
triangle(50, "solid", "red")
```

1	What is the name of this function?	
2	What did the expression evaluate to?	
3	How many arguments does `triangle` expect?	
4	What data type does the `triangle` function produce? (Numbers? Strings? Booleans?)	

Catching Bugs

The following lines of code are all BUGGY! Read the code and the error messages to identify the mistake.

5) `triangle(20, "solid" "red")`

Pyret didn't understand your program around
triangle(20, "solid" **"red"**)

Can you spot the mistake? ______________________________

6) `triangle(20, "solid")`

This application expression errored:
triangle (*20* , *"solid"*)
2 arguments were passed to the **operator** . The **operator** evaluated to a function accepting 3 parameters. An application expression expects the number of parameters and *arguments* to be the same.

Can you spot the mistake? ______________________________

7) `triangle(20, 10, "solid", "red")`

This application expression errored:
triangle (20, 10, "solid", *"red"*)`
4 arguments were passed to the **operator** . The **operator** evaluated to a function accepting 3 parameters. An application expression expects the number of parameters and *arguments* to be the same.

Can you spot the mistake? ______________________________

8) `triangle (20, "solid", "red")`

Pyret thinks this code is probably a function call:
triangle *(20, "solid", "red")*
Function calls must not have space between the **function expression** and the *arguments* .

Can you spot the mistake? ______________________________

Domain and Range

My Definition

Facts & Characteristics

Domain

Examples

Non-examples

My Definition

Facts & Characteristics

Range

Examples

Non-examples

Practicing Contracts: Domain & Range

Consider the following contract:

```
is-beach-weather :: Number, String -> Boolean
```

1) What is the **Name** of this function? ______________________

2) How many arguments are in this function's **Domain**? ______________________

3) What is the **type** of this function's **first argument**? ______________________

4) What is the **type** of this function's **second argument**? ______________________

5) What is the **Range** of this function? ______________________

6) Circle the expression below that shows the correct application of this function, based on its contract.

A. `is-beach-weather(70, 90)`

B. `is-beach-weather(80, 100, "cloudy")`

C. `is-beach-weather("sunny", 90)`

D. `is-beach-weather(90, "stormy weather")`

Consider the following contract:

```
cylinder :: Number, Number, String -> Image
```

7) What is the **Name** of this function? ______________________

8) How may arguments are in this function's **Domain**? ______________________

9) What is the **type** of this function's **first argument**? ______________________

10) What is the **type** of this function's **second argument**? ______________________

11) What is the **type** of this function's **third argument**? ______________________

12) What is the **Range** of this function? ______________________

13) Circle the expression below that shows the correct application of this function, based on its contract.

A. `cylinder("red", 10, 60)`

B. `cylinder(30, "green")`

C. `cylinder(10, 25, "blue")`

D. `cylinder(14, "orange", 25)`

Matching Expressions and Contracts

Match the contract (left) with the expression described by the function being used (right).

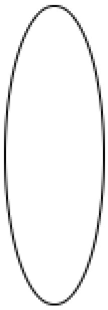
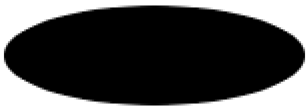
Contract		Expression
`# make-id :: String, Number -> Image`	1	A `make-id("Savannah", "Lopez", 32)`
`# make-id :: String, Number, String -> Image`	2	B `make-id("Pilar", 17)`
`# make-id :: String -> Image`	3	C `make-id("Akemi", 39, "red")`
`# make-id :: String, String -> Image`	4	D `make-id("Raïssa", "McCracken")`
`# make-id :: String, String, Number -> Image`	5	E `make-id("von Einsiedel")`

Contract		Expression
`# is-capital :: String, String -> Boolean`	6	A `show-pop("Juneau", "AK", 31848)`
`# is-capital :: String, String, String -> Boolean`	7	B `show-pop("San Juan", 395426)`
`# show-pop :: String, Number -> Image`	8	C `is-capital("Accra", "Ghana")`
`# show-pop :: String, String, Number -> Image`	9	D `show-pop(3751351, "Oklahoma")`
`# show-pop :: Number, String -> Number`	10	E `is-capital("Albany", "NY", "USA")`

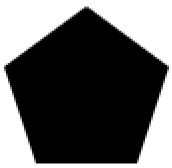
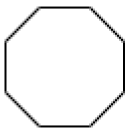
Using Contracts

Use the contracts to write expressions to generate images similar to those pictured.

`ellipse :: Number, Number, String, String -> Image`

	
	
What changes with the first number?	
What about the shape changes with the second Number?	
Write an expression using `ellipse` to produce a circle.	

`regular-polygon :: Number, Number, String, String -> Image`

	
	
What changes with the first Number?	
What about the shape changes with the second Number?	
Use `regular-polygon` to write an expression for a square!	
How would you describe a **regular polygon** to a friend?	

Triangle Contracts

1) What kind of triangle does the `triangle` function produce? ____________________

There are lots of other kinds of triangles! And Pyret has lots of other functions that make triangles!

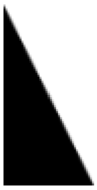
```
triangle :: (size:: Number, style :: String, color :: String) -> Image
right-triangle :: (base::Number, height::Number, style::String, color::String) -> Image
isosceles-triangle :: (leg::Number, angle::Number, style::String, color::String) -> Image
```

2) Why do you think `triangle` only needs one number, while `right-triangle` and `isosceles-triangle` need two numbers and `triangle-sas` needs three?

____

____

3) Write `right-triangle` expressions for the images below. *One argument for each should be* `100`.

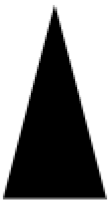


____

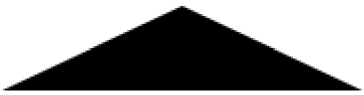
____

4) What do you think the numbers in `right-triangle` represent? ____________________

5) Write `isosceles-triangle` expressions for the images below. *1 argument for each should be* `100`.



____



____

6) What do you think the numbers in `isosceles-triangle` represent?

____

7) Write 2 expressions that would build **right-isosceles** triangles. Use `right-triangle` for one expression and `isosceles-triangle` for the other expression.

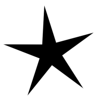
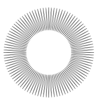
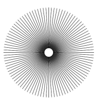
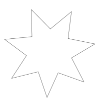
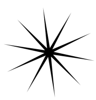
____

____

Radial Star

```
radial-star :: (
    points :: Number,
    inner-radius :: Number,
    full-radius :: Number,
    style :: String,
    color :: String
) -> Image
```

Using the detailed contract above, match each image to the expression that describes it.

Image			Expression
	1	A	`radial-star(5, 50, 200, "solid", "black")`
	2	B	`radial-star(7, 100, 200, "solid", "black")`
	3	C	`radial-star(7, 100, 200, "outline", "black")`
	4	D	`radial-star(10, 150, 200, "solid", "black")`
	5	E	`radial-star(10, 20, 200, "solid", "black")`
	6	F	`radial-star(100, 20, 200, "solid", "black")`
	7	G	`radial-star(100, 100, 200, "outline", "black")`

What’s on your mind?

Displaying Categorical Data

Data Scientists use **displays** to visualize data. You've probably seen some of these charts, graphs and plots yourselves! When it comes to displaying **Categorical Data**, there are two displays that are especially useful.

1. **Bar charts** show the *count or percentage* of rows in each category.
   - Bar charts provide a visual representation of the frequency of values in a categorical column.
   - Bar charts have a bar for every category in a column.
   - The more rows in a category, the taller the bar.
   - Bars in a bar chart can be show in *any order*, without changing the meaning of the chart. However, bars are usually shown in some sensible order (bars for the number of orders for different t-shirt sizes might be presented in order of smallest to largest shirt).
2. **Pie charts** show the *percentage* of rows in each category.
   - Pie charts provide a visual representation of the relative frequency of values in a categorical column.
   - Pie charts have a slice for every category in a column.
   - The more rows in a category, the larger the slice.
   - Slices in a pie chart can be shown in *any order*, without changing the meaning of the chart. However, slices are usually shown in some sensible order (e.g. slices might be shown in alphabetical order or from the smallest to largest slice).

Exploring Displays

Using your Contracts page and the Animals Starter File, make each type of display below in pyret. Then sketch the displays and answer the questions. Be sure to add examples of the code you use to your contracts page!

Pie Charts	Bar Charts
Sketch a pie chart here.	Sketch a bar chart here.
Displays _1_ column(s) of _categorical_ data. What does this display tell us? ____________________ ____________________ ____________________ ____________________	Displays ___ column(s) of __________ data. What does this display tell us? ____________________ ____________________ ____________________ ____________________
Box Plots	**Histograms**
Sketch a box plot here	Sketch a histogram here
Displays ___ column(s) of __________ data. What do you think this display tells us? ____________________ ____________________ ____________________ ____________________	Displays ___ column(s) of __________ data. What do you think this display tells us? ____________________ ____________________ ____________________ ____________________

(More) Exploring Displays

For each type of display, fill in the information below.

Scatter Plot	Linear Regression Plot
Sketch a scatter plot here.	Sketch a linear regression plot here.
Displays ___ column(s) of ________________ data. What do you think this display tells us? ______________________________ ______________________________ ______________________________ ______________________________	Displays ___ column(s) of ________________ data. What do you think this display tells us? ______________________________ ______________________________ ______________________________ ______________________________

What’s on your mind?

Data Displays and Lookups

Data scientists use data visualizations to gain better insights into their data, and to communicate their findings with others. Making a display requires answering three questions:

1. **What data** is being displayed? This could be "a random sample of 2000 people", "every animal from the shelter", or "students aged 14-17".
2. **What variables** are being explored? Are we looking at the `species` column? The number of `kilograms` that an animal weighs? Searching for a relationship between a person's `income` and their `height` ?
3. **What display** is being used, given the variables being explored? If it's a quantitative variable, we might use a histogram or box plot. If it's categorical, we could use a pie or bar chart. If it's two quantitative variables, we probably want a scatter plot.

Defining Values, Looking up Rows and Columns

We can define names for values in Pyret, the same way we do in math:

```
name = "Flannery"
age = 16
logo = star(50, "solid", "red")
```

When **looking up a data Row** from a Table, programmers use the `row-n` method. This method takes a single number as its input, which tells the computer which Row we want. *Note: Rows are numbered starting at zero!*
For example:

```
sasha   = animals-table.row-n(0)  # define sasha to be the first row
mittens = animals-table.row-n(2)  # define mittens to be the third row
```

When **looking up a column** from a Row, programmers use square brackets and the name of the column they want.
For example:

```
animals-table.row-n(0)["age"]  # look up the age in the 1st row
mittens["species"]             # look up the species in the third row
```

Throughout the rest of the workbook, we will sometimes refer to animalA and animalB as rows from the table.

```
animalA = animals-table.row-n(4)
animalB = animals-table.row-n(13)
```

What Display Goes with Which Data?

Match the Display with the description of the data being plotted. Some descriptions may go with more than one display!

Pie Charts	**1**	**A**	1 column of Quantitative Data
Bar Charts	**2**		
Histograms	**3**	**B**	2 columns of Quantitative Data
Box Plots	**4**		
Scatter Plots	**5**	**C**	1 column of Categorical Data

Data Displays

Fill in the tables below, then use Pyret to make the following displays. Record the code you used.
The first column has been filled in for you.

1) A `pie-chart` showing the `species` of animals from the shelter.

Which Rows?	Which Column(s)?	What Display?
All the animals		

code:

2) A `bar-chart` showing the `sex` of animals from the shelter.

Which Rows?	Which Column(s)?	What Display?
All the animals		

code:

3) A `histogram` of the number of `pounds` that animals weigh.

Which Rows?	Which Column(s)?	What Display?
All the animals		

code:

4) A `box-plot` of the number of `pounds` that animals weigh.

Which Rows?	Which Column(s)?	What Display?
All the animals		

code:

5) A `scatter-plot`, using the animals' `species` as the labels, `age` as the x-axis, and `pounds` as the y-axis.

Which Rows?	Which Column(s)?	What Display?
All the animals		

code:

6) A `scatter-plot`, using the animals' `name` as the labels, `pounds` as the x-axis, and `weeks` as the y-axis.

Which Rows?	Which Column(s)?	What Display?
All the animals		

code:

Lookup Questions

The table below represents four pets at an animal shelter:

`pets-table`

name	sex	age	pounds
"Toggle"	"female"	3	48
"Fritz"	"male"	4	92
"Nori"	"female"	6	35.3
"Maple"	"female"	3	51.6

1) *Match* each Lookup Question (left) to the code that will give the answer (right).

"How much does Maple weigh?"	**1**	**A**	`pets-table.row-n(3)`
"Which is the last row in the table?	**2**	**B**	`pets-table.row-n(2)["name"]`
"What is Fritz's sex?"	**3**	**C**	`pets-table.row-n(1)["sex"]`
"What's the third animal's name?"	**4**	**D**	`pets-table.row-n(3)["age"]`
"How much does Nori weigh?"	**5**	**E**	`pets-table.row-n(3)["pounds"]`
"How old is Maple?"	**6**	**F**	`pets-table.row-n(0)`
"What is Toggle's sex?"	**7**	**G**	`pets-table.row-n(2)["pounds"]`
"What is the first row in the table?"	**8**	**H**	`pets-table.row-n(0)["sex"]`

2) Fill in the blanks (left) with code that will produce the value (right).

a.	*pets-table.row-n(3)["name"]*	"Maple"
b.	______________________	"male"
c.	______________________	4
d.	______________________	48
e.	______________________	"Nori"

What’s on your mind?

Defining Row Functions & Using Table Methods

Methods are special functions that are attached to pieces of data. We use them to manipulate Tables.

- In this course, the methods we'll be using are
  - `row-n` - consumes an index (starting with zero!) and produces a row from a table
  - `order-by` - consumes the name of a column and a Boolean value to determine if that table should be sorted by that column in ascending order
  - `filter` - consumes a *Boolean-producing function*, and produces a table containing only rows for which the function returns `true`
  - `build-column` - consumes the name of a new column, and a function that produces the values in that column for each Row
- Unlike functions, methods can't be used alone. They have a "secret" argument, which is the data they are attached to. They are written as part of that data, separated by a dot. For example:

  ```
  shapes.row-n(2)
  ```

- Contracts for methods are different from other functions. They include the type of the data as part of their names. For example:

  ```
  <table>.row-n :: (index :: Number) -> Row
  ```

Reading Row and Function Definitions

Make sure you've opened the [Table Methods Starter File] on your computer.

1	What name is being defined on line 15?	
2	How many columns are listed here?	
3	What name is being defined on line 22?	
4	Is `cat-row` a Number, String, Image or Row?	
5	Type `cat-row` into the Interactions Area. What do you get?	

6) On line 27, define `dog-row` . After clicking "Run", type `dog-row` into the Interactions Area and make sure it's a dog! Do the same for `old-row` and `unfixed-row` .

7	A Contract for a function is written on line 39. What is its name?	
8	What is its Domain?	
9	What is its Range?	
10	What other functions are defined here?	

11) Lines 41-42 **define a new function!** What does this function do?

____

____

Exploring Row and Function Definitions

Make sure you've opened the Table Methods Starter File on your computer.

1	Evaluate `is-dog(dog-row)`. What do you get?	
2	Evaluate `is-cat(cat-row)`. What do you get?	
3	Evaluate `is-cat(dog-row)`. What do you get?	
4	Evaluate `is-dog(dog-row)`. What do you get?	
5	Evaluate `is-dog(cat-row)`. What do you get?	
6	What does `is-cat` do?	
7	What does `lookup-fixed` do?	
8	What does `is-old` do?	
9	What does `kilos` do?	
10	What does `nametag` do?	

11) Find the Contract for `image-scatter-plot` in your Contracts page, and discuss the Domain as a group.

12) In the Interactions Area, type `image-scatter-plot(animals-table, "pounds", "weeks", nametag)`. What do you get?

---

13) Change the definition of `nametag` to produce text with a different color.

14) Change the definition of `nametag` to produce text with a different size.

15) Change the definition of `nametag` to produce text using the animal's `species`, instead of their `name`.

16) Change the definition of `nametag` to produce text using the animal's `age` as the size of the text.

Defining Functions

Functions can be viewed in *multiple representations*. You already know one of them: ***Contracts***, which specify the Name, Domain, and Range of a function. Contracts are a way of thinking of functions as a *mapping* between one set of data and another. For example, a mapping from Numbers to Strings:

```
f :: Number -> String
```

Another way to view functions is with ***Examples***. Examples are essentially input-output tables, showing what the function would do for a specific input:

In our programming langauge, we focus on the last two columns and write them as code:

```
examples:
  f(1) is 1 + 2
  f(2) is 2 + 2
  f(3) is 3 + 2
  f(4) is 4 + 2
end
```

Finally, we write a formal **function definition** ourselves. The pattern in the Examples becomes *abstract* (or "general"), replacing the inputs with ***variables***. In the example below, the same definition is written in both math and code:

$f(x) = x + 2$

```
fun f(x): x + 2 end
```

Look for connections between these three representations!

- The function name is always the same, whether looking at the Contract, Examples, or Definition.
- The number of inputs in the Examples is always the same as the number of types in the Domain, which is always the same as the number of variables in the Definition.
- The "what the function does" pattern in the Examples is almost the same in the Definition, but with specific inputs replaced by variables.

Matching Examples and Definitions (Math)

Look at each set of examples on the left and circle what is changing from one example to the next.
Then, *match* the examples on the left to the definitions on the right.

Examples: **Functions:**

x	$f(x)$
1	2×1
2	2×2
3	2×3

1

x	$f(x)$
15	$15 - 3$
25	$25 - 3$
35	$35 - 3$

2

x	$f(x)$
10	$10 + 2$
15	$15 + 2$
20	$20 + 2$

3

x	$f(x)$
0	$3(0) - 2$
1	$3(1) - 2$
2	$3(2) - 2$

4

x	$f(x)$
10	$2(10) + 1$
20	$2(20) + 1$
30	$2(30) + 1$

5

A $f(x) = x - 3$

B $f(x) = 2x$

C $f(x) = 2x + 1$

D $f(x) = 3x - 2$

E $f(x) = x + 2$

Matching Examples and Function Definitions

Highlight the variables in `gt` and label them with the word "size".

```
examples:
  gt(20) is
    triangle(20, "solid", "green")
  gt(45) is
    triangle(45, "solid", "green")
end
```

```
fun gt(size): triangle(size, "solid", "green") end
```

Highlight and label the variables in the example lists below. Then, using `gt` as a model, match the examples to their corresponding function definitions.

Examples		Definition
`examples:` `  f("solid") is` `    circle(8, "solid", "red")` `  f("outline") is` `    circle(8, "outline", "red")` `end`	1	A `fun f(s): star(s, "outline", "red") end`
`examples:` `  f(2) is 2 + 2` `  f(4) is 4 + 4` `  f(5) is 5 + 5` `end`	2	B `fun f(num): num + num end`
`examples:` `  f("red") is circle(7, "solid", "red")` `  f("teal") is` `    circle(7, "solid", "teal")` `end`	3	C `fun f(c): star(9, "solid", c) end`
`examples:` `  f("red") is star(9, "solid", "red")` `  f("grey") is star(9, "solid", "grey")` `  f("pink") is star(9, "solid", "pink")` `end`	4	D `fun f(s): circle(8, s, "red") end`
`examples:` `  f(3) is star(3, "outline", "red")` `  f(8) is star(8, "outline", "red")` `end`	5	E `fun f(c): circle(7, "solid", c) end`

Matching Examples and Contracts

Match each set of examples (left) with the contract that best describes it(right).

<table>
<tr><th>Examples</th><th></th><th></th><th>Contract</th></tr>
<tr><td><pre>examples:
  f(5) is 5 / 2
  f(9) is 9 / 2
  f(24) is 24 / 2
end</pre></td><td>1</td><td>A</td><td><code># f :: Number -> Number</code></td></tr>
<tr><td><pre>examples:
  f(1) is
    rectangle(1, 1, "outline", "red")
  f(6) is
    rectangle(6, 6, "outline", "red")
end</pre></td><td>2</td><td>B</td><td><code># f :: String -> Image</code></td></tr>
<tr><td><pre>examples:
  f("pink", 5) is
    star(5, "solid", "pink")
  f("blue", 8) is
    star(8, "solid", "blue")
end</pre></td><td>3</td><td>C</td><td><code># f :: Number -> Image</code></td></tr>
<tr><td><pre>examples:
  f("Hi!") is text("Hi!", 50, "red")
  f("Ciao!") is text("Ciao!", 50, "red")
end</pre></td><td>4</td><td>D</td><td><code># f :: Number, String -> Image</code></td></tr>
<tr><td><pre>examples:
  f(5, "outline") is
    star(5, "outline", "yellow")
  f(5, "solid") is
    star(5, "solid", "yellow")
end</pre></td><td>5</td><td>E</td><td><code># f :: String, Number -> Image</code></td></tr>
</table>

Contracts, Examples & Definitions

gt

Directions : Define a function called `gt`, which makes solid green triangles of whatever size we want.

Every contract has three parts...

\# gt:: Number -> Image

function name *domain* *range*

Write some examples, then circle and label what changes...

examples:

```
gt( 10 ) is triangle(10, "solid", "green")
```
function name *input(s)* *what the function produces*

```
gt( 20 ) is triangle(20, "solid", "green")
```
function name *input(s)* *what the function produces*

end

Write the definition, giving variable names to all your input values...

```
fun gt( size ):
```
function name *variable(s)*

```
triangle(size, "solid", "green")
```
what the function does with those variable(s)

end

bc

Directions : Define a function called `bc`, which makes solid blue circles of whatever radius we want.

Every contract has three parts...

\# ______ :: ______ -> ______

function name *domain* *range*

Write some examples, then circle and label what changes...

examples:

______ (______) **is** ______

function name *input(s)* *what the function produces*

______ (______) **is** ______

function name *input(s)* *what the function produces*

end

Write the definition, giving variable names to all your input values...

fun ______ (______):

function name *variable(s)*

______

what the function does with those variable(s)

end

What’s on your mind?

The Design Recipe

Functions have multiple representations (e.g. - Contracts, Examples, and Definition), and each of these representations shows us a particular part of how the function should behave. By using these representations in a particular order - called the ***Design Recipe*** - we can build lots of functions, check our work, and document our thinking!

Contract and Purpose Statement

The first step in the Design Recipe is to write the Contract. This means we have to be able to answer three questions:

- What is the **Name** of the function we are defining?
- What is the **Domain** of that function? (When dealing with Table Functions, the Domain is always `Row`)
- What is the **Range** of the function? (What is the type of the output?)

The Purpose Statement is a way of adding detail to the Contract, using plain human language. A good Purpose Statement should always explain:

- What the input *represents*. (Is it Animals? Schools? States?)
- What the output *represents*. (Pounds? True or false?)
- All the information necessary to go from input to output.

It's important to start with this representation, because it's the least detailed. If we can't answer *these* questions, we shouldn't start writing code!

Examples

The second step is work through some concrete examples, making sure that we know exactly what the function will do.
The goal of the Examples step is to *find the pattern* that represents what the function does. Sometimes we have to start by just focusing on what the answer should be. Suppose `animalA` is a lizard animal, and `animalB` isn't. We can imagine the answers for an `is-lizard` to be...

```
examples:
    is-lizard(animalA) is true
    is-lizard(animalB) is false
end
```

But what work do we have to do to check if an animal is a lizard? (1) We **look up** the "species" column, and (2) ask if the value is equal to "lizard". We can write both of these steps in code, finishing the examples:

```
examples:
    is-lizard(animalA) is animalA["species"] == "lizard"
    is-lizard(animalB) is animalB["species"] == "lizard"
end
```

(And sometimes we can go straight to showing our work, doing the whole thing in one step!)
Once we see the pattern, we can *circle and label what changes*. In this case, only the animal itself changes!

Definition

The final step in the Design Recipe is to take the pattern from our examples and *generalize it* to work with any input.
Once again, our previous step is a huge help: we can simply **copy everything that stays the same**, and replace the part that changes with the label we used:

```
fun is-lizard(r): r["species"] == "lizard" end
```

The Design Recipe - Compute

For the word problems below, assume `dog-row`, `cat-row`, `young-row` and `old-row` are already defined as data rows.

Directions: Define a function called `is-cat`, which consumes a `Row` of the animals table and *computes* whether the animal is a cat.

Contract and Purpose Statement

Every contract has three parts...

\# is-cat:: Row -> Boolean

function name | domain | range

\# Consumes an animal, and computes whether the species equals "cat"

what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

is-cat(dog-row) **is** ______

function name | input(s) | what the function produces

______ (______) **is** ______

function name | input(s) | what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun is-cat(r):

function name | variable(s)

r["species"] == "cat"

what the function does with those variable(s)

end

Directions: Define a function called `is-young`, which consumes a Row of the animals table and *computes* whether it is less than four years old.

Contract and Purpose Statement

Every contract has three parts...

\# ______ :: ______ -> ______

function name | domain | range

\# ______

what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

______ (______) **is** ______

function name | input(s) | what the function produces

______ (______) **is** ______

function name | input(s) | what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun ______ (______):

function name | variable(s)

______

what the function does with those variable(s)

end

The Design Recipe - Lookup

For the word problems below, assume `fixed` and `unfixed` are already defined as data rows.

Directions : Define a function called `lookup-fixed` , which looks up whether or not an animal is fixed.

Contract and Purpose Statement

Every contract has three parts...

__________ :: __________ -> __________
function name | domain | range

____
what does the function do?

Examples

Write some examples, then circle and label what changes...

```
examples:
  lookup-fixed(  fixed-row    )  is  fixed-row["fixed"]
  lookup-fixed(  unfixed-row  )  is  unfixed-row["fixed"]
end
```
function name | input(s) | what the function produces

Definition

Write the definition, giving variable names to all your input values...

```
fun lookup-fixed(  r  ):
  r["fixed"]
end
```
function name | variable(s) | what the function does with those variable(s)

Directions : Define a function called `lookup-name` , which consumes a Row of the animals table and looks up the name of that animal.

Contract and Purpose Statement

Every contract has three parts...

#lookup-name:: Row -> String
function name | domain | range

#Consumes an animal, and looks up the name
what does the function do?

Examples

Write some examples, then circle and label what changes...

```
examples:
  ________( ________ ) is ________
  ________( ________ ) is ________
end
```
function name | input(s) | what the function produces

Definition

Write the definition, giving variable names to all your input values...

```
fun ________( ________ ):
  ________
end
```
function name | variable(s) | what the function does with those variable(s)

What’s on your mind?

Method Chaining

Method chaining allows us to apply multiple methods with less code.

For example, instead of using multiple definitions, like this:

```
with-labels = animals-table.build-column("labels", nametag)
cats = with-labels.filter(is-cat)
cats.order-by("age", true)
```

We can use method-chaining to write it all on one line, like this:

```
animals-table.build-column("labels", nametag).filter(is-cat).order-by("age", true)
```

Order Matters! The methods are applied in the order they appear. For example, trying to order a table by a column that hasn't been built will result in an error.

The Design Recipe: is-dog / is-female

For the word problems below, assume you have `animalA` and `animalB` defined in your code.

Directions: Define a function called `is-dog`, which consumes a `Row` of the animals table and *computes* whether the animal is a dog.

Contract and Purpose Statement

Every contract has three parts...

\# is-dog:: Row -> Boolean

function name | domain | range

\# Consumes an animal, and computes whether the species == "dog"

what does the function do?

Examples

Write some examples, then circle and label what changes...

```
examples:
  is-dog(  animalA  )  is  animalA["species"] == "dog"
  is-dog(  animalB  )  is  ____________________
end
```

function name | input(s) | what the function produces

Definition

Write the definition, giving variable names to all your input values...

```
fun is-dog(  r  ):
  r["species"] == "dog"
end
```

function name | variable(s) | what the function does with those variable(s)

Directions: Define a function called `is-female`, which consumes a Row of the animals table and returns true if the animal is female.

Contract and Purpose Statement

Every contract has three parts...

\# ____________ :: ____________ -> ____________

function name | domain | range

\# ____________

what does the function do?

Examples

Write some examples, then circle and label what changes...

```
examples:
  ________( ________ ) is ________
  ________( ________ ) is ________
end
```

function name | input(s) | what the function produces

Definition

Write the definition, giving variable names to all your input values...

```
fun ________( ________ ):
  ________
end
```

function name | variable(s) | what the function does with those variable(s)

The Design Recipe: is-old / name-has-s

For the word problems below, assume you have `animalA` and `animalB` defined in your code.

Directions : Define a function called `is-old` , which consumes a Row of the animals table and *computes* whether it is more than 12 years old.

Contract and Purpose Statement

Every contract has three parts...

__________ :: __________ -> __________

function name domain range

____

what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

__________ (__________) **is** __________

function name input(s) what the function produces

__________ (__________) **is** __________

function name input(s) what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun __________ (__________):

function name variable(s)

____

what the function does with those variable(s)

end

Directions : Define a function called `name-has-s` , which returns true if an animal's name contains the letter "s"

Contract and Purpose Statement

Every contract has three parts...

__________ :: __________ -> __________

function name domain range

____

what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

__________ (__________) **is** __________

function name input(s) what the function produces

__________ (__________) **is** __________

function name input(s) what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun `name-has-s(` `r`):

function name variable(s)

`string-contains(r["name"],  "s")`

what the function does with those variable(s)

end

Chaining Methods

You have the following functions defined below (read them *carefully!*):

```
fun is-fixed(r): r["fixed"]                     end
fun is-young(r): r["age"] < 4                   end
fun nametag(r):  text(r["name"], 20, "red") end
```

The table `t` below represents four animals from the shelter:

name	sex	age	fixed	pounds
`"Toggle"`	`"female"`	`3`	`true`	`48`
`"Fritz"`	`"male"`	`4`	`true`	`92`
`"Nori"`	`"female"`	`6`	`true`	`35.3`
`"Maple"`	`"female"`	`3`	`true`	`51.6`

Match each Pyret expression (left) to the description of what it does (right).

`t.order-by("age", true)`	1	A	Produces a table containing only Toggle and Maple
`t.filter(is-fixed)`	2	B	Produces a table of only young, fixed animals
`t.build-column("sticker", nametag)`	3	C	Produces a table, sorted youngest-to-oldest
`t.filter(is-young)`	4	D	Produces a table with an extra column, named "sticker"
`t.filter(is-young)` `  .filter(is-fixed)`	5	E	Produces a table containing Maple and Toggle, in that order
`t.filter(is-young)` `  .order-by("pounds", false)`	6	F	Produces a table containing the same four animals
`t.build-column("label", nametag)` `  .order-by("age", true)`	7	G	Won't run: will produce an error
`t.order-by("sx", false)`	8	H	Produces a table with an extra "label" column, sorted youngest-to-oldest

Chaining Methods 2: Order Matters

You have the following functions defined below (read them *carefully!*):

```
fun is-female(r): r["sex"] == "female"  end
fun kilograms(r): r["pounds"] / 2.2     end
fun is-heavy(r):  r["kilos"] > 25       end
```

The table `t` below represents four animals from the shelter:

name	sex	age	fixed	pounds
"Toggle"	"female"	3	true	48
"Fritz"	"male"	4	true	92
"Nori"	"female"	6	true	35.3
"Maple"	"female"	3	true	51.6

Match each Pyret expression (left) to the description of what it does (right). **Note: one description might match multiple expressions!**

`t.order-by("kilos", true)`	1	A	Produces a table containing Toggle, Nori and Maple, with an extra column showing their weight in kilograms
`t.filter(is-female)` `.build-column("kilos", kilograms)`	2	B	Produces a table containing Maple, Nori and Toggle (in that order)
`t.build-column("kilos", kilograms)` `.filter(is-heavy)`	3	C	Produces a table containing only Fritz.
`t.filter(is-heavy)` `.build-column("kilos", kilograms)`	4	D	Won't run: will produce an error
`t.build-column("kilos", kilograms)` `.filter(is-heavy)` `.order-by("sex", true)`	5	E	Produces a table containing only Fritz, with two extra columns.
`t.build-column("female", is-female)` `.build-column("kilos", kilograms)` `.filter(is-heavy)`	6	F	Produces a table containing Maple and Fritz

What’s on your mind?

Mood Generator

1) Open the Mood Generator Starter File, and read throught the code you find there. This code contains new programming that you haven't seen yet! Take a moment to list everything you Notice, and then everything you Wonder…

Notice	Wonder

2) Add another line of code to the definition, so that `mood("mad")` produces the *same* emoji as `mood("angry")`.

3) Add **another example** to the **`examples:`** section for "laughing", using the appropriate emoji. (To bring up the emojis on your computer, type `Cmd-Ctrl-Space` on a Mac, or `Windows-Period` on Windows 10)

4) Come up with some new moods, and add them to the code. Make sure you include **`examples:`**!

5) In your own words, how do if-expressions work in Pyret? Write your answer below.

6) Write down at least 2 ways you could use if-expressions when analyzing the Animals Dataset.

Word Problem: species-color

Directions : We want to generate a custom dot for our `image-scatter-plot` , such that every species gets a unique color. Write a function called `species-color` , which takes in a Row from the animals table and returns a solid, 5px circle using a color you've chosen.

Contract and Purpose Statement

Every contract has three parts...

\# _______________ :: ____ -> _______________
function name | domain | range

\# ____
what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

_______________ (_______________) **is** _______________________________
function name | input(s) | what the function produces

_______________ (_______________) **is** _______________________________
function name | input(s) | what the function produces

_______________ (_______________) **is** _______________________________
function name | input(s) | what the function produces

_______________ (_______________) **is** _______________________________
function name | input(s) | what the function produces

_______________ (_______________) **is** _______________________________
function name | input(s) | what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun _______________ (_______________________):
function name | variable(s)

____
what the function does with those variable(s)

____
what the function does with those variable(s)

____
what the function does with those variable(s)

____
what the function does with those variable(s)

____
what the function does with those variable(s)

end

Randomness and Sample Size

Computer Scientists may take **samples** that are subsets of a dataset. If their sample is well chosen, they can use it to test if their code does what it's supposed to do. However, choosing a good sample can be tricky!

Random Samples are a subset of a population in which each member of the subset has an equal chance of being chosen. A random sample is intended to be a representative subset of the population. The larger the random sample, the more closely it will represent the population and the better our inferences about the population will tend to be.

Grouped Samples are a subset of a population in which each member of the subset was chosen for a specific reason. For example, we might want to look at the difference in trends between two groups ("Is the age of a dog a bigger factor in adoption time v. the age of a cat?"). This would require making grouped samples of *just the dogs* and *just the cats.*

Sampling and Inference

1) Evaluate the `big-animals-table` in the Interactions Area. This is the *complete* population of animals from the shelter! Below is a true statement about that population:

> The population is 47.7% fixed and 52.3% unfixed.

Type each of the following lines into the Interactions Area and hit "Enter".

```
random-rows(big-animals-table, 10)
random-rows(big-animals-table, 40)
```

2) What do you get?

____

3) What is the contract for `random-rows`?

____

4) What does the `random-rows` function do?

____

____

5) In the Definitions Area, define `small-sample` and `large-sample` to be these two random samples.

6) Make a `pie-chart` for the animals in each sample, showing percentages of fixed and unfixed.

- The percentage of fixed animals in the entire population is **47.7%**.
- The percentage of fixed animals in `small-sample` is ____________.
- The percentage of fixed animals in `large-sample` is ____________.

7) Make a `pie-chart` for the animals in each sample, showing percentages for each species.

- The percentage of tarantulas in the entire population is **roughly 5%**.
- The percentage of tarantulas in `small-sample` is ____________.
- The percentage of tarantulas in `large-sample` is ____________.

8) Click "Run" to direct the computer to generate a different set of random samples of these sizes. Make a new `pie-chart` for each sample, showing percentages for each species.

- The percentage of tarantulas in the entire population is **roughly 5%**.
- The percentage of tarantulas in `small-sample` is ____________.
- The percentage of tarantulas in `large-sample` is ____________.

9) Which repeated sample gave us a more accurate inference about the whole population? Why?

____

____

Grouped Samples from the Animals Dataset

Use method chaining to define the **grouped samples** below, using the helper functions that you've already defined: `is-old`, `is-young`, `is-cat`, `is-dog`, `is-female`, `is-fixed`, and `name-has-s`. We've given you the solution for the first sample, to get you started.

	Subset	The code to define that subset
1	Kittens	`kittens = animals-table.filter(is-cat).filter(is-young)`
2	Puppies	
3	Fixed Cats	
4	Cats with "s" in their name	
5	Old Dogs	
6	Fixed Animals	
7	Old Female Cats	
8	Fixed Kittens	
9	Fixed Female Dogs	
10	Old Fixed Female Cats	

Displaying Data

Fill in the tables below, then use Pyret to make the following displays. Record the code you used. The first table has been filled in for you.

1) A `bar-chart` showing how many puppies are fixed or not.

What Rows?	Which Column(s)?	What Display?
puppies	*fixed*	*bar-chart*

code: `bar-chart(animals-table.filter(is-dog).filter(is-young), "fixed")`

2) A `pie-chart` showing how many heavy dogs are fixed or not.

What Rows?	Which Column(s)?	What Display?

code: ______________________________

3) A `histogram` of the number of `weeks` it takes for a random sample of animals to be adopted.

What Rows?	Which Column(s)?	What Display?

code: ______________________________

4) A `box-plot` of the number of `pounds` that kittens weigh.

What Rows?	Which Column(s)?	What Display?

code: ______________________________

5) A `scatter-plot` of a random sample using `species` as the labels, `age` as the x-axis, and `weeks` as the y-axis.

What Rows?	Which Column(s)?	What Display?

code: ______________________________

6) Describe **your own grouped sample** here, and fill in the table below.

____

What Rows?	Which Column(s)?	What Display?

code: ______________________________

What's on your mind?

Choosing Your Dataset

When selecting a dataset to explore, *pick something that matters to you!* You'll be working with this data for a while, so you don't want to pick something at random just to get it done.

When choosing a dataset, it's a good idea to consider a few factors:

1. Is it **interesting**? This should be data you are curious about, that answers questions you'd want to ask. Pick a dataset you're genuinely interested in, so that you can explore questions that matter to you!
2. Is it **relevant**? Does this data impact you in any way? Are there questions you have about the dataset that mean something to you or someone you know? Pick a dataset that deals with something personally relevant to you!
3. Is it **familiar**? You wouldn't be able to make samples of the Animals Dataset properly if you didn't know that some animals are much bigger or longer-lived than others. Pick a dataset you know about, so you can use your expertise to deepen your analysis!

My Dataset

I chose to work with the ______________________________ dataset, which contains ________ data rows.
For each question, **can it be answered by this dataset?** Make sure you have at least two questions that can be answered, and at least one that cannot.

What do you NOTICE?	What do you WONDER?	Answered by this dataset?
		Yes *No*
		Yes *No*
		Yes *No*
		Yes *No*
		Yes *No*
		Yes *No*

Choose two columns to describe below

1) ____________________ (column name), which contains ________________ (categorical/quantitative) data. Example values from this column include:

____

2) ____________________ (column name), which contains ________________ (categorical/quantitative) data. Example values from this column include:

____

Samples from My Dataset

Think back to when we defined grouped samples from the Animals Table, like "puppies", "old cats", etc. What grouped samples would be useful for *your* dataset? List a few of these in the first column.

Then, for each one, what function will identify if a row `r` is in the subset?

Grouped Sample	A function that returns true if a row `r` is in the subset
	`fun ____________________(r):` `end`
	`fun ____________________(r):` `end`
	`fun ____________________(r):` `end`
	`fun ____________________(r):` `end`
	`fun ____________________(r):` `end`

The Design Recipe

Write helper functions for **your** dataset, which you can use to define subsets. Since all helper functions will consume Rows, their Domains have already been filled in for you.

Directions : Define a function called ______________________, which consumes a Row of the ______________________ table and produces ______________________.

Contract and Purpose Statement

Every contract has three parts...

______________ :: Row -> ______________
function name / domain / range

____
what does the function do?

Examples

Write some examples, then circle and label what changes...

```
examples:
  ______________(______________) is ______________
  ______________(______________) is ______________
end
```

function name / input(s) / what the function produces

Definition

Write the definition, giving variable names to all your input values...

```
fun ______________(______________):
  ____________________________________________
end
```

function name / variable(s) / what the function does with those variable(s)

Directions : Define a function called ______________________, which consumes a Row of the ______________________ table and produces ______________________.

Contract and Purpose Statement

Every contract has three parts...

______________ :: Row -> ______________
function name / domain / range

____
what does the function do?

Examples

Write some examples, then circle and label what changes...

```
examples:
  ______________(______________) is ______________
  ______________(______________) is ______________
end
```

function name / input(s) / what the function produces

Definition

Write the definition, giving variable names to all your input values...

```
fun ______________(______________):
  ____________________________________________
end
```

function name / variable(s) / what the function does with those variable(s)

The Design Recipe

Write helper functions for **your** dataset, which you can use to define subsets. Since all helper functions will consume Rows, their Domains have already been filled in for you.

Directions: Define a function called ______________________, which consumes a Row of the ______________________ table and produces ______________________.

Contract and Purpose Statement

Every contract has three parts…

______________ :: ______ Row ______ -> ______
function name — domain — range

____
what does the function do?

Examples

Write some examples, then circle and label what changes…

```
examples:
  ______________ ( ______________ ) is ______________________
  function name     input(s)           what the function produces
  ______________ ( ______________ ) is ______________________
  function name     input(s)           what the function produces
end
```

Definition

Write the definition, giving variable names to all your input values…

```
fun ______________ ( ______________________ ):
    function name       variable(s)
  ______________________________________________
  what the function does with those variable(s)
end
```

Directions: Define a function called ______________________, which consumes a Row of the ______________________ table and produces ______________________.

Contract and Purpose Statement

Every contract has three parts…

______________ :: ______ Row ______ -> ______
function name — domain — range

____
what does the function do?

Examples

Write some examples, then circle and label what changes…

```
examples:
  ______________ ( ______________ ) is ______________________
  function name     input(s)           what the function produces
  ______________ ( ______________ ) is ______________________
  function name     input(s)           what the function produces
end
```

Definition

Write the definition, giving variable names to all your input values…

```
fun ______________ ( ______________________ ):
    function name       variable(s)
  ______________________________________________
  what the function does with those variable(s)
end
```

What's on your mind?

Histograms

To best understand histograms, it's helpful to contrast them first with bar charts.

Bar charts show the number of rows belonging to a given category. The more rows in each category, the taller the bar.

- *Bar charts provide a visual representation of the frequency of values in a **categorical** column.*
- There's no strict numerical way to order these bars, but **sometimes there's an order** that makes sense. For example, bars for the sales of different t-shirt sizes might be presented in order of smallest to largest shirt.

Histograms show the number of rows that fall within certain intervals, or "bins", on a horizontal axis. The more rows that fall within a particular "bin", the taller the bar.

- *Histograms provide a visual representation of the frequencies (or relative frequencies) of values in a **quantitative** column.*
- Quantitative data **can always be ordered**, so the bars of a histogram always progress from smallest (on the left) to largest (on the right).
- 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 shape of the dataset. Choosing a good bin size can take some trial and error!

The **shape** of a dataset tells us which values are more or less common.

- In a **symmetric** dataset, values are just as likely to occur a certain distance above the mean as below the mean.
- Some extreme values may be far greater or far lower the other values in a dataset. These extreme values are called **outliers**.
- A dataset that is **skewed left** and/or has low outliers has a few values that are unusually low. The histogram for a skewed left dataset has a few data points that are stretched out to the left (lower) end of the x-axis.
- A dataset that is **skewed right** and/or high outliers means there are a few values that are unusually high. The histogram for a skewed right dataset has a few data points that are stretched out to the right (higher) end of the x-axis.
- One way to visualize the difference between a histogram of data that is **skewed left** or **skewed right** is to think about the lengths of our toes on our left and right feet. Much like a histogram that is "skewed left", our left feet have smaller toes on the left and a bigger toe on the right. Our right feet have the big toe on the left and smaller toes on the right, more closely resembling the shape of a histogram of "skewed right" data.

The Design Recipe

For the word problems below, assume you have `animalA` and `animalB` defined in your code.

Directions : Define a function called `kilos` , which consumes a Row of the animals table and divides the pounds column by 2.2 to *compute* the animal's weight in kilograms.

Contract and Purpose Statement

Every contract has three parts...

__________ :: (r :: Row) -> __________

function name | domain | range

____

what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

__________ (__________) **is** __________

function name | input(s) | what the function produces

__________ (__________) **is** __________

function name | input(s) | what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun __________ (__________):

function name | variable(s)

____

what the function does with those variable(s)

end

Directions : Define a function called `smart-dot` , which consumes a Row of the animals table and *computes* the image of a solid red circle using the animal's `pounds` as the radius.

Contract and Purpose Statement

Every contract has three parts...

smart-dot:: __________ -> Image

function name | domain | range

Consumes an animal, and computes a solid red circle using the weight in pounds as the radius

what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

smart-dot("animalA") **is** __________

function name | input(s) | what the function produces

__________ (__________) **is** __________

function name | input(s) | what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun __________ (__________):

function name | variable(s)

____

what the function does with those variable(s)

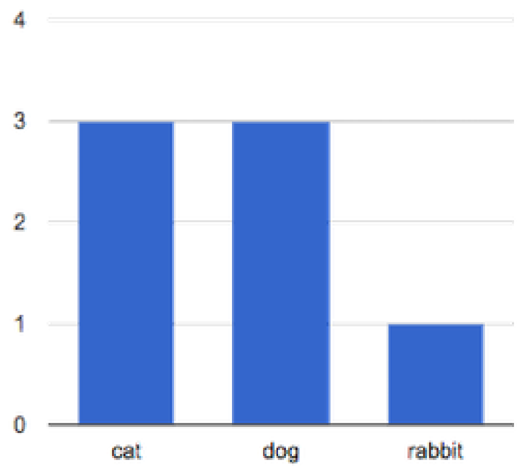
end

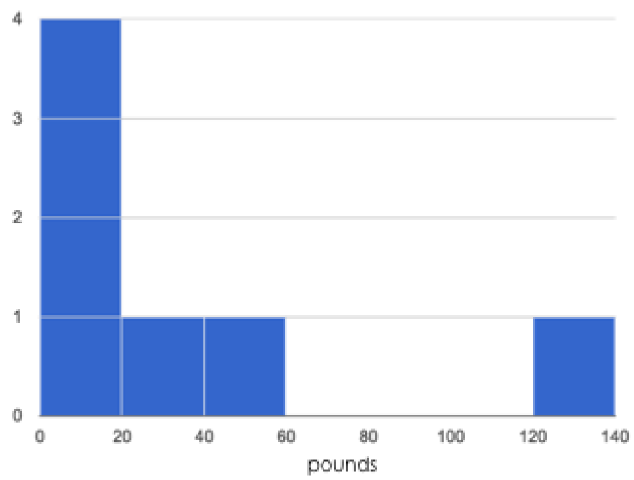
Summarizing Columns

name	species	age	pounds
"Sasha"	"cat"	1	6.5
"Boo-boo"	"dog"	11	12.3
"Felix"	"cat"	16	9.2
"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 in the table above?	
2	How many dogs are there?	
3	How many animals weigh between 0-20 pounds?	
4	How many animals weigh between 20-40 pounds?	
5	Are there more animals weighing 40-60 than 60-140 pounds?	

The charts below are both based on this table. What is similar about them? What is different?





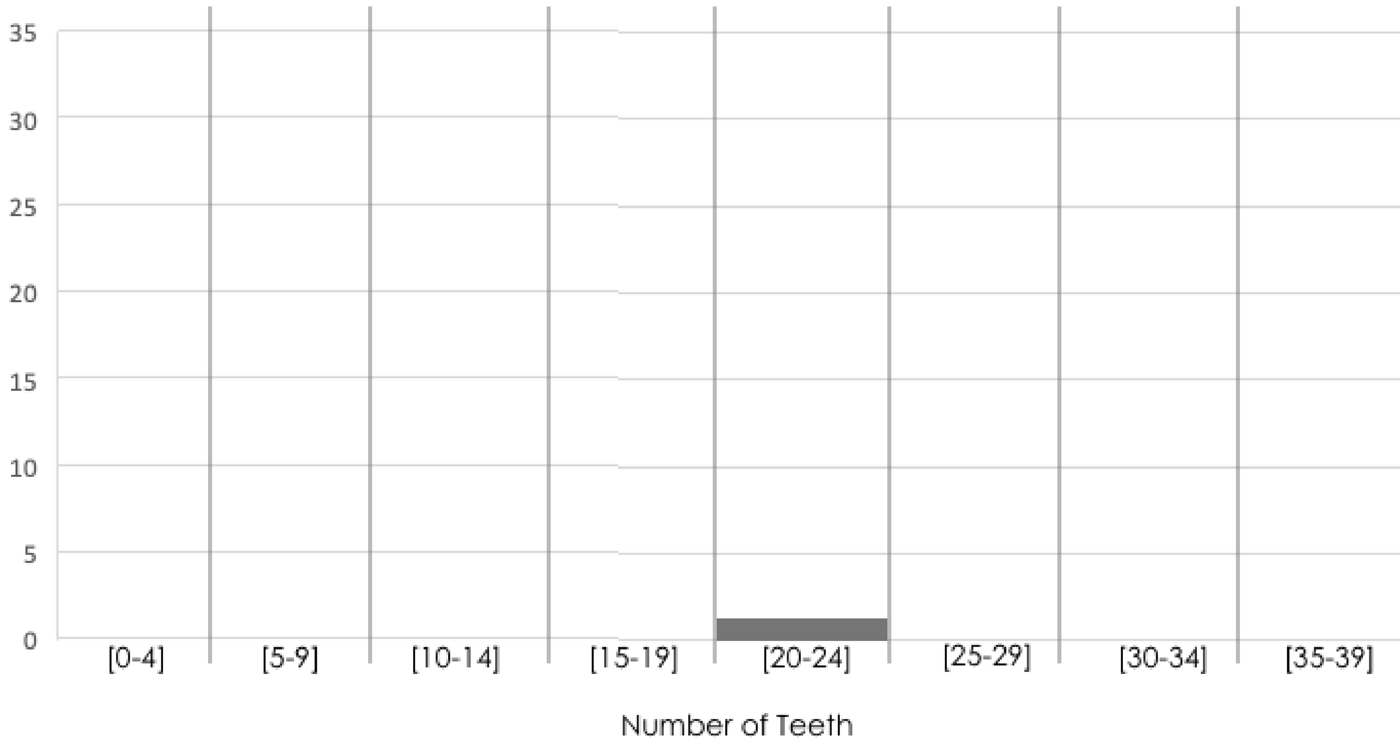
Similarities	Differences

Making Histograms

Suppose we have a dataset for a group of 50 adults, showing the number of teeth each person has:

Number of teeth	Count
0	5
22	1
26	1
27	1
28	4
29	3
30	5
31	3
32	27

Draw a histogram for the table in the space below. For each row, find which interval (or "bin") on the x-axis represents the right number of teeth. Then fill in the box so that the height of the box is equal to the *sum of the counts* that fit into that interval. One of the intervals has been completed for you.

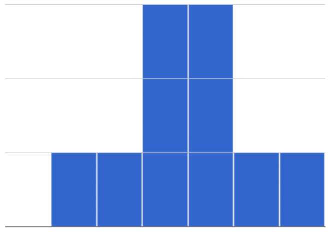


Reading Histograms

Students watched 5 videos, and rated them on a scale of 1 to 10. While the **average score** for every video is the same (5.5), the **shapes** of the ratings distributions were very different! *Match* the summary description (left) with the *shape* of the histogram of student ratings (right). For each histogram, **the x-axis is the score, and the y-axis is the number of students who gave it that score**. These axes are intentionally unlabeled - focusing on the *shape* is what matters here!

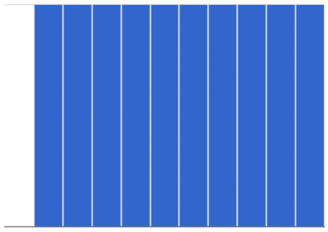
Most of the students were fine with the video, but a couple of them gave it an unusually low rating. **1**

A



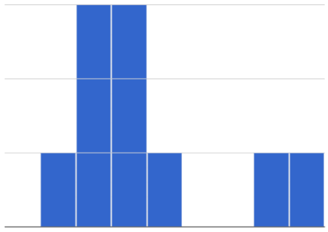
Most of the students were okay with the video, but a couple students gave it an unusually high rating. **2**

B



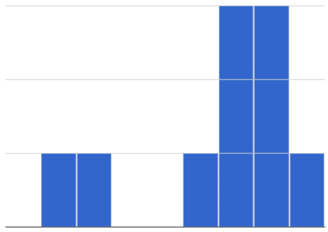
Students tended to give the video an average rating, and they weren't likely to stray far from the average. **3**

C



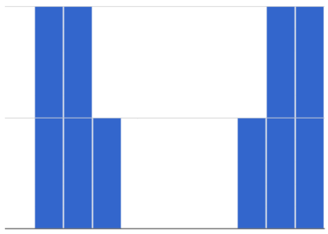
Students either really liked or really disliked the video. **4**

D



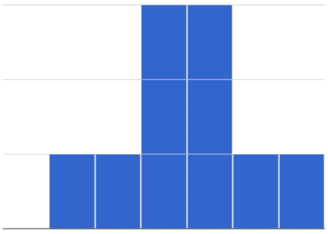
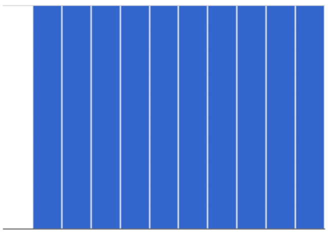
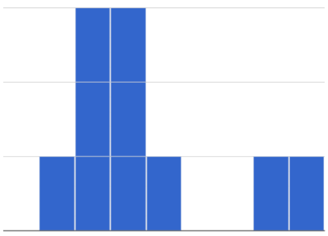
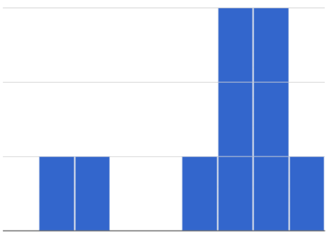
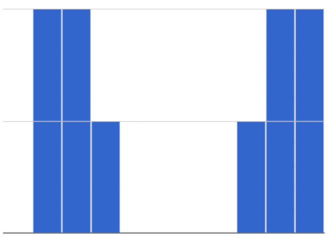
Reactions to the video were all over the place: high ratings and low ratings and inbetween ratings were all equally likely. **5**

E



Identifying Shape - Histograms

Describe the shape of histograms on the left in complete sentences, using vocabulary like "Skewed Left", "Skewed Right", or "Symmetric".

1		
2		
3		
4		
5		

The Shape of the Animals Dataset

Describe two histograms made from columns of the animals dataset.

1) Make a histogram, showing the distribution of ______pounds______ for
column in your dataset

______animals from the shelter______.
your subset, e.g., "fixed dogs from the shelter"

2) Make another histogram, showing the distribution of ____________ for
column in your dataset

____________.
your subset, e.g., "fixed dogs from the shelter"

3) How would you describe the shape of these histograms?

____

What do you NOTICE?	What do you WONDER?

The Spread of My Dataset

Describe two of the histograms you made from your dataset.

1) I made a histogram, showing the distribution of ______________________________ for
column in your dataset

____.
your subset, e.g., "fixed dogs from the shelter"

2) I made a histogram, showing the distribution of ______________________________ for
column in your dataset

____.
your subset, e.g., "fixed dogs from the shelter"

3) How would you describe the shape of these histograms?

____

What do you NOTICE about these displays?	What do you WONDER about these displays?

What’s on your mind?

Measures of Center and Spread

There are three ways to measure the **center** of a dataset, to summarize a whole column of quantitative data using just one number:

- The **mean** of a dataset is the average of all the numbers.
- The **median** of a dataset is a value that is smaller than half the dataset, and larger than the other half. In an ordered list the median will either be the middle number or the average of the two middle numbers.
- The **mode(s)** of a dataset is the value (or values) occurring most often. When all of the values occur equally often, a dataset has no mode.

In a **symmetric** dataset, values are just as likely to occur a certain distance above the mean as below the mean, and the median and mean are usually close together.

When a dataset is asymmetric, the median is a more decriptive measure of center than the median.

- A dataset with **left skew**, and/or low outliers, has a few values that are unusually low, pulling the mean *below* the median.
- A dataset with **right skew**, and/or high outliers, means there are a few values that are unusually high, pulling the mean *above* the median.

When a dataset contains a small number of values, the mode may be the most descriptive measure of center.

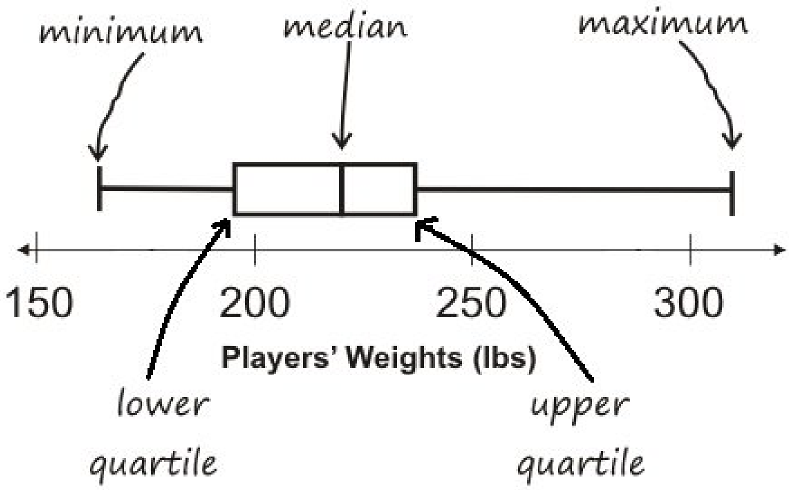
Data Scientists can also measure the **spread** of a dataset using a **five-number summary**:

- The **minimum** – the lowest value in the dataset
- The **first, or “lower” quartile (Q1)** – the middle of the lower half of values, which separates the lowest quarter from the next smallest quarter
- The **second quartile (Q2)** – the middle value, which separates the entire dataset into “top” and “bottom” halves
- The **third, or “upper” quartile (Q3)** – the middle of the higher half of values which separates the second highest quarter from the highest quarter
- The **maximum** – the largest value in the dataset

Measures of Center and Spread (continued)

The **five-number summary** can be used to draw a **box plot**.

- Each of the four sections of the box plot contains 25% of the data. *If the values are distributed evenly across the range, the four sections of the box plot will be equal in width.* Uneven distributions will show up as differently-sized sections of a box plot.
- The left **whisker** extends from the minimum to Q1.
- The **box**, or **interquartile range**, extends from Q1 to Q3. It is divided into 2 parts by the **median**. Each of those parts contains 25% of the data, so the whole box contains the central 50% of the data.
- The right **whisker** extends from Q3 to the maximum.



The box plot above, for example, tells us that:

- The minimum weight is about 165 pounds. The median weight is about 220 pounds. The maximum weight is about 310 pounds.
    - 1/4 of the players weigh roughly between 165 and 195 pounds
    - 1/4 of the players weigh roughly between 195 and 220 pounds
    - 1/4 of the players weigh roughly between 220 and 235 pounds
    - 1/4 of the players weigh roughly between 235 and 310 pounds
    - 50% of the players weigh roughly between 165 and 220 pounds
    - 50% of the players weigh roughly between 195 and 235 pounds
    - 50% of the players weigh roughly between 220 and 310 pounds
- The densest concentration of players' weights is between 220 and 235 pounds.
- Because the widest section of the box plot is between 235 and 310 pounds, we understand that the weights of the heaviest 25% fall across a wider span than the others. 310 may be an outlier, the weights of the players weighing between 235 pounds and 310 pound could be evenly distributed across the range, or all of the players weighing over 235 pounds may weigh around 310 pounds.

Summarizing Columns in the Animals Dataset

Find the measures of center and spread to summarize the ______pounds______ column of the Animals Table.
Be sure to add examples to your Contracts page as you work.

Measures of Center

The three measures of center for this column are:

Mean (Average)	Median	Mode(s)

Since the mean is ______________ compared to the median, this suggests the shape is
[higher/lower/about equal]

____.
[skewed right (or high outliers) / skewed left (or low outliers) / symmetric]

Measures of Spread

My five-number summary is:

Minimum	Q1	Median	Q3	Maximum

Displaying Center and Spread with a Box Plot

Draw a box plot from this summary on the number line below.
Be sure to label the number line with consistent intervals.

From this summary and box plot, I conclude:

Interpreting Spread

Consider the following dataset, representing the annual income of ten people.
All numbers represent *thousands of dollars* (so `14` means "$14,000"):

```
60, 10, 21, 180, 14, 20, 45, 35, 45, 170
```

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

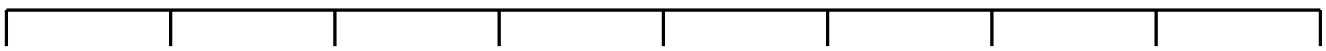
2) In the table below, compute the **measures of center** for this dataset.

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.

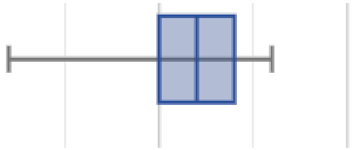


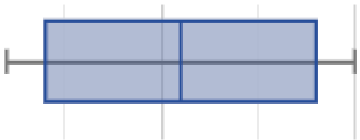
5) The following statements are *correct* ... but misleading. Write down the reason why.

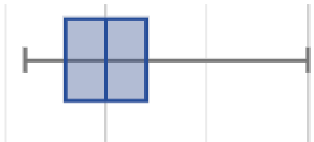
Statement	Why it's misleading
"They're rich! The average person makes $60k dollars!"	
"It's a middle-income list: the most common salary is $45k/yr!"	
"This group is very low-income, the most common salary range is from $10k-$25k!"	

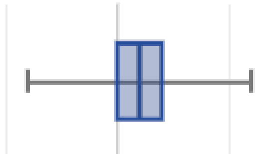
Identifying Shape - Box Plots

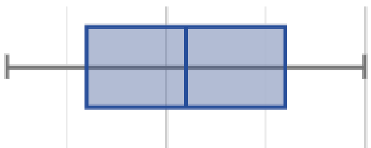
Describe the shape of box plots on the left in complete sentences, using vocabulary like "Skewed Left", "Skewed Right" or "Symmetric".

1 

2 

3 

4 

5 

Shape of My Dataset

Find the measures of center and spread to summarize a column of your dataset.

The column I chose to summarize is ______________________________.

Measures of Center

The three measures of center for this column are:

Mean (Average)	Median	Mode(s)

Since the mean is ______________ compared to the median, this suggests the shape is
[higher/lower/about equal]

____.
[skewed right (or high outliers) / skewed left (or low outliers) / symmetric]

Measures of Spread

My five-number summary is:

Minimum	Q1	Q2 (Median)	Q3	Maximum

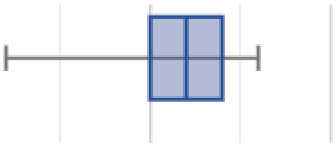
Displaying Center and Spread with a Box Plot

Draw a box plot from this summary on the number line below.
Be sure to label the number line with consistent intervals.

From this summary and box plot, I conclude:

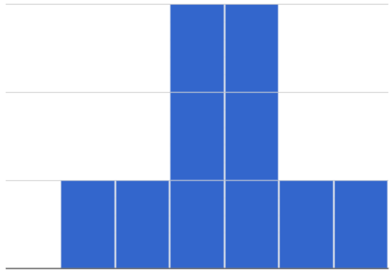
Matching Box-Plots to Histograms

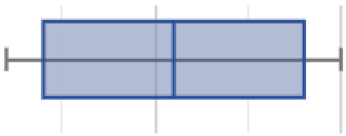
Students watched 5 videos, and rated them on a scale of 1 to 10. For each video, their ratings were used to generate box-plots and histograms. Match the box-plot to the histogram that displays the same data.*



1

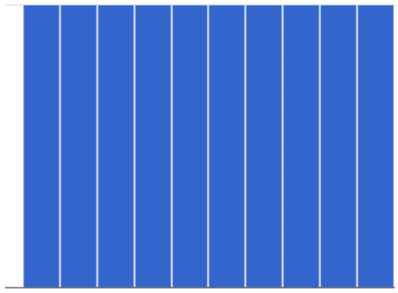
A

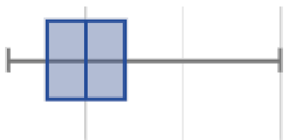




2

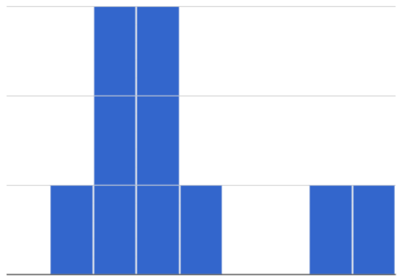
B

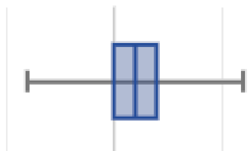




3

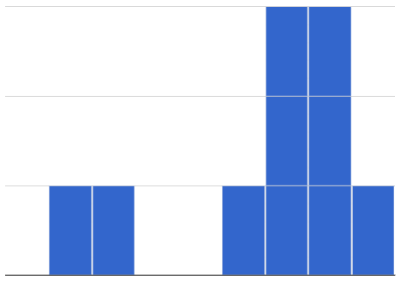
C

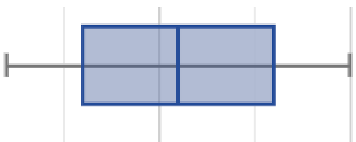




4

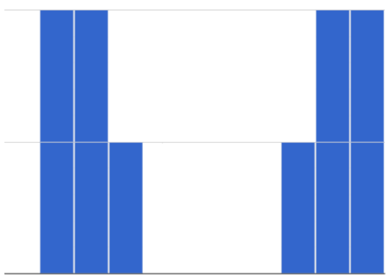
D





5

E



What’s on your mind?

"Trust, but verify ..."

A "helpful" Data Scientist gives you access to the following function:

```
# fixed-cats :: Table -> Table
# consumes a table of animals, and produces a table containing only cats that have been fixed, sorted from youngest-to-oldest
```

You can use the function, *but you can't see the code for it!* **How do you know if you can trust their code?**

- You could make a *verification subset* that contains one of every species, and make sure that the function filters out everything but cats.
- You could make sure this subset has multiple cats not already ordered of youngest-to-oldest, and make sure the function puts them in the right order.

1) What other qualities would this subset need to have?

2) Create your verification subset! In the space below, list the name of each animal in your subset.

Name

“Trust, but verify…” (2)

A “helpful” Data Scientist gives you access to the following function:

```
# old-dogs-nametags :: Table -> Table
# consumes a table of animals, and produces a table containing only dogs 5 years or older, with an extra column showing their name in red
```

You can use the function, *but you can’t see the code for it!* **How do you know if you can trust their code?**

1) What qualities would a verification subset need to have?

2) Create your verification subset! In the space below, list the name and index of each animal in your subset.

Name

What’s on your mind?

Scatter Plots

Scatter Plots can be used to show a 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 straight line, it suggests that a linear relationship exists between those two columns. A number called a **correlation** can be used to summarize this relationship.
- r is the name of the **correlation statistic**. The r-value will always fall between −1 and +1. The sign tells us whether the correlation is positive or negative. Distance from 0 tells us the strength of the correlation.
  - −1 or +1 are the strongest possible negative and possible correlations.
  - 0 means no correlation.
- The correlation is **positive** if the point cloud slopes up as it goes farther to the right. This means larger y-values tend to go with larger x-values. It is **negative** if it slopes down as it goes farther to the right.
- If the points are tightly clustered around a line, it is a **strong** correlation. That means knowing the x-value gives us a pretty good idea of the y-value. If they are loosely scattered it is a **weak** correlation, and the y-value doesn’t depend much on the x-value.
- Points that are far above or below the cloud of points in a scatter plot are called **outliers**.
- We graphically summarize this relationship by drawing a straight line through the data cloud, so that the vertical distance between the line and all the points taken together is as small as possible. This line is called the **line of best fit** and allows us to predict y-values based on x-values.

(Dis)Proving a Claim

"Smaller animals get adopted faster because they're easier to care for."

Do you agree? If so, why?

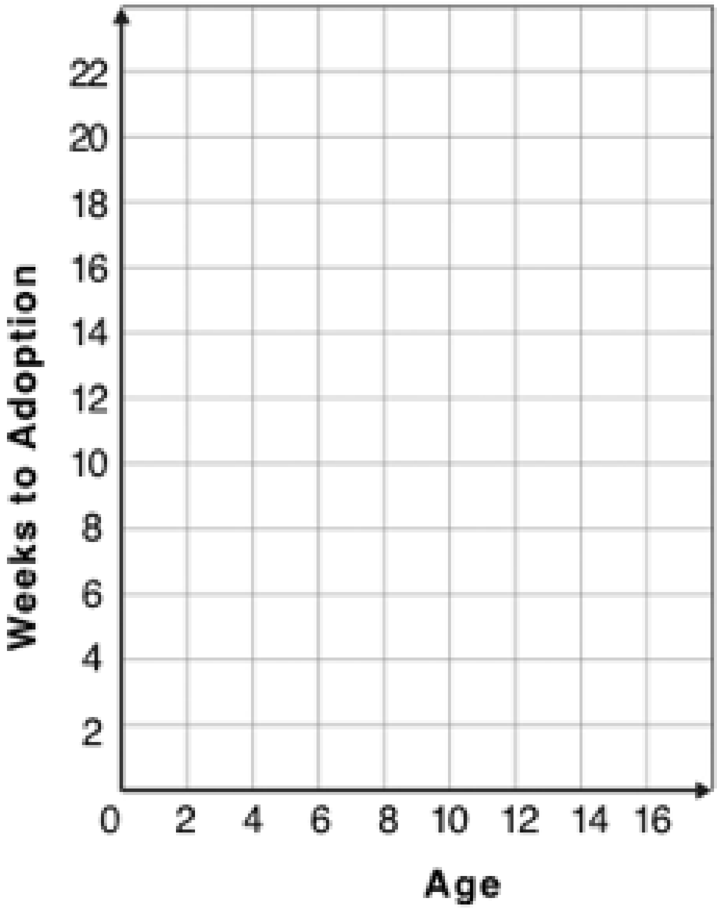
I hypothesize …

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

Creating a Scatter Plot

1. **For each row in the Sample Table on the left, add a point to the scatter plot on the right** . 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 go up or down as age increases to the right?
   - **Draw a cloud around all the points, and a line around which the cloud appears to be centered**
3. Does the line slope upwards or downwards? ______________________________
4. Are the points tightly clustered around the line or loosely scattered? ______________________________

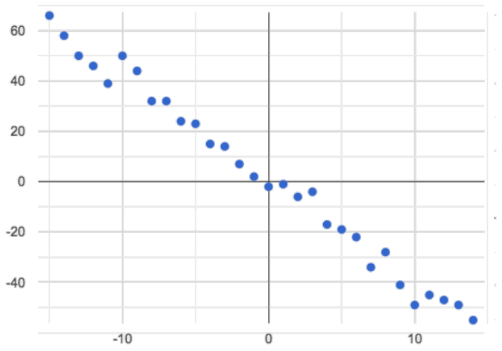
name	species	age	weeks
"Sasha"	"cat"	1	3
"Boo-boo"	"dog"	11	5
"Felix"	"cat"	16	4
"Buddy"	"lizard"	2	24
"Nori"	"dog"	6	9
"Wade"	"cat"	1	2
"Nibblet"	"rabbit"	6	12
"Maple"	"dog"	3	2



Identifying Form, Direction and Strength

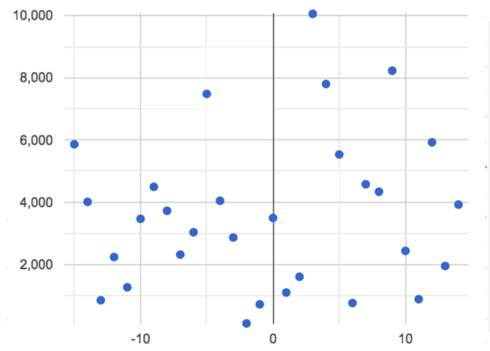
Can you identify the Form, Direction, & Strength of these displays? **Note:** If the form is non-linear, we shouldn't report direction - a curve may rise and then fall

A



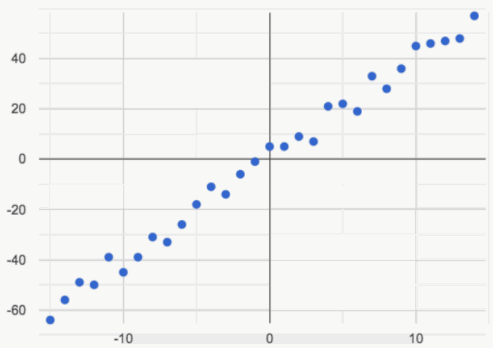
Form : Linear Non-Linear None
Direction : Positive Negative None
Strength : Strong Weak None

B



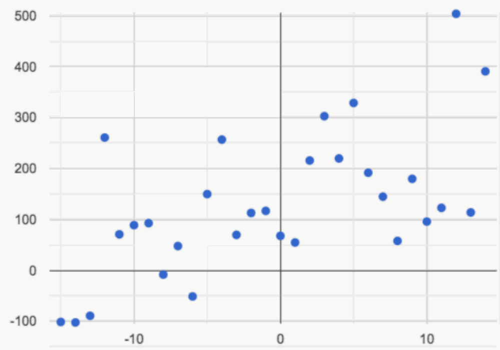
Form : Linear Non-Linear None
Direction : Positive Negative None
Strength : Strong Weak None

C



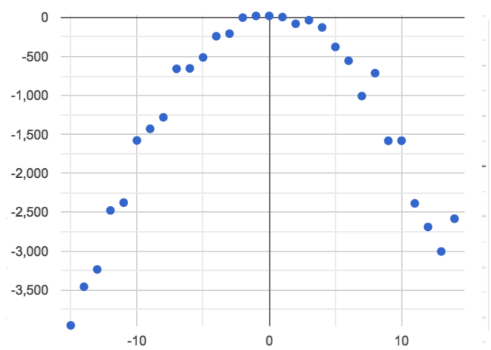
Form : Linear Non-Linear None
Direction : Positive Negative None
Strength : Strong Weak None

D



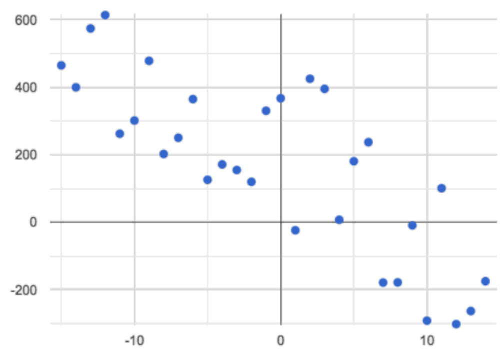
Form : Linear Non-Linear None
Direction : Positive Negative None
Strength : Strong Weak None

E



Form : Linear Non-Linear None
Direction : Positive Negative None
Strength : Strong Weak None

F

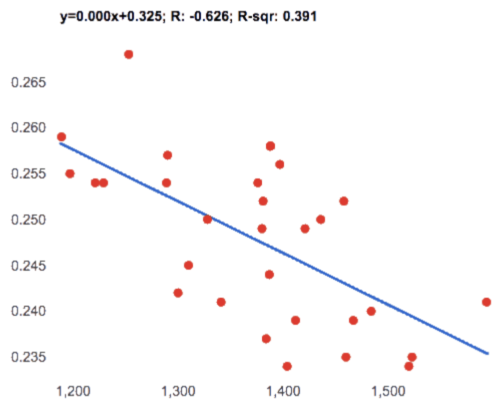


Form : Linear Non-Linear None
Direction : Positive Negative None
Strength : Strong Weak None

Identifying Form and r-Values

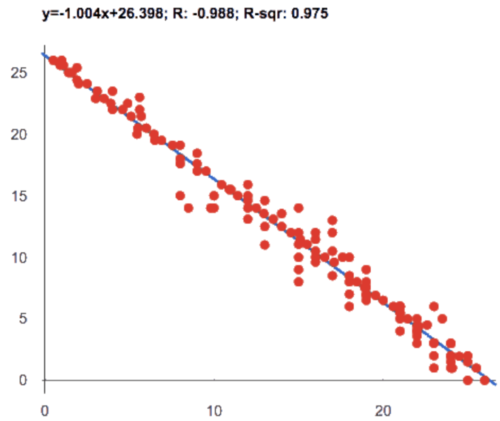
Can you identify the Form and r-Values of these displays?
If the form is linear, approximate the r-value to express Direction and Strength.
Reminder: An r-value close to -1 is a strong negative relationship, an r-value close to 0 is weak, and an r-value close to +1 is a strong positive! If the relationship's strength is moderate, the r-value will be closer to -0.5 or +0.5.

A



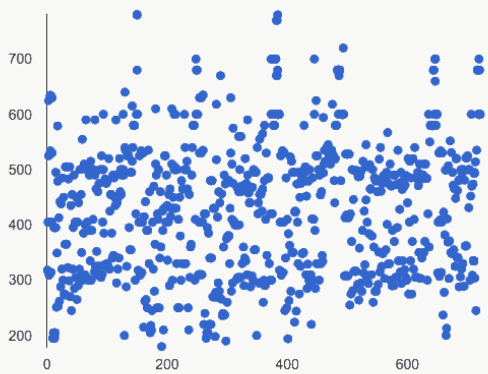
Form :
r close to :

B



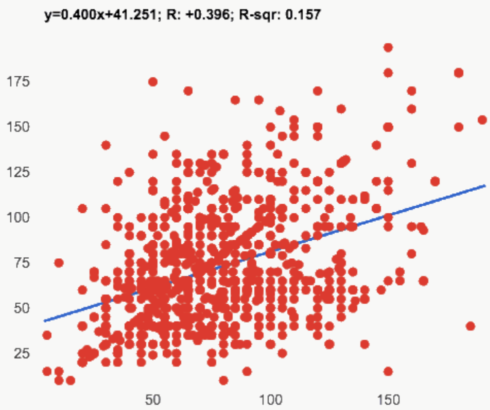
Form :
r close to :

C



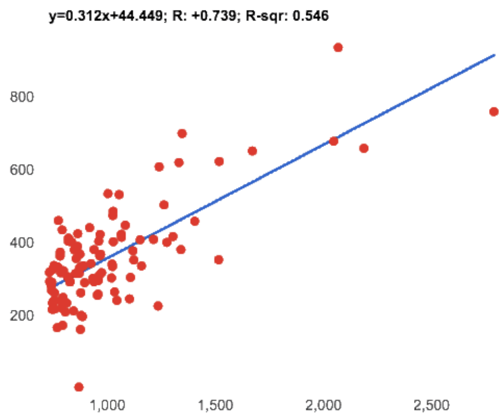
Form :
r close to :

D



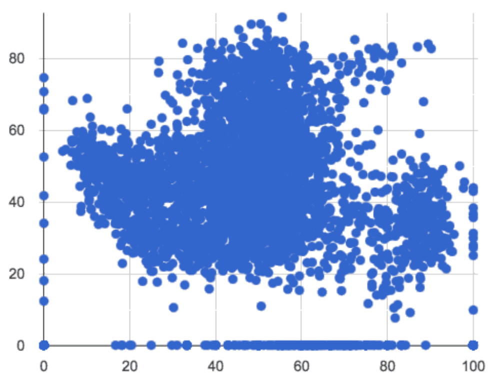
Form :
r close to :

E



Form :
r close to :

F



Form :
r close to :

Correlations in My Dataset

1) There may be a correlation between ______________________ and ______________________.
column column

I think it is a ______________________, ______________________ correlation,
strong/weak positive/negative

because ____.

It might be stronger if I looked at ______________________________.
a sample or extension of my data

---

2) There may be a correlation between ______________________ and ______________________.
column column

I think it is a ______________________, ______________________ correlation,
strong/weak positive/negative

because ____.

It might be stronger if I looked at ______________________________.
a sample or extension of my data

---

3) There may be a correlation between ______________________ and ______________________.
column column

I think it is a ______________________, ______________________ correlation,
strong/weak positive/negative

because ____.

It might be stronger if I looked at ______________________________.
a sample or extension of my data

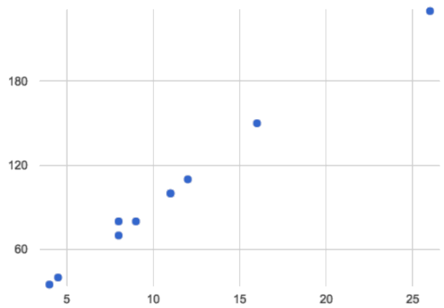
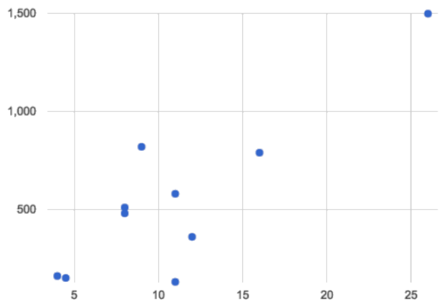
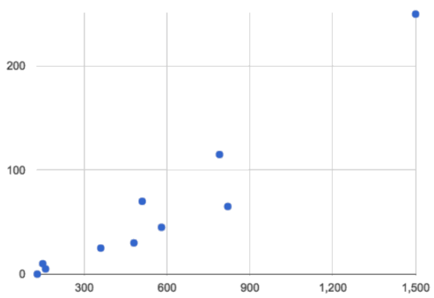
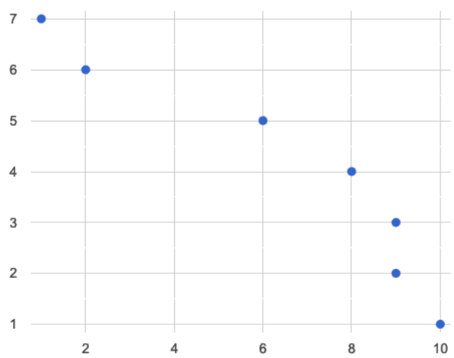
Computing Relationships

Linear Regression is a way of computing the **line of best fit**, which minimizes the *sum of the squares* of the vertical distances from the points to the line. Calculating the slope and intercept of this line is a task best left to computing or statistical software.

- * **Slope** provides us with the easiest summary to grasp: it's how much we predict the ***response variable*** (plotted on the y-axis) will increase or decrease for each unit that the ***explanatory variable*** (plotted on the x-axis) increases.
- **Correlation is not causation!** Correlation only suggests that two column variables are related, but does not tell us if one causes the other. For example, hot days are correlated with people running their air conditioners, but air conditioners do not cause hot days!
- **Sample size matters!** The number of data values is also relevant. We'd be more convinced of a positive relationship in general between cat age and time to adoption if a correlation of +0.57 were based on 50 cats instead of 5.

Drawing Predictors

For each of the scatter plots below, draw a **predictor line** that seems like the best fit. Describe the correlation in terms of Direction and Strength, then estimate the r-value as being close to -1, -0.5, 0, +0.5, or +1.

A		**Direction :** Positive Negative None **Strength :** Strong Weak **r :** ________
B		**Direction :** Positive Negative None **Strength :** Strong Weak **r :** ________
C		**Direction :** Positive Negative None **Strength :** Strong Weak **r :** ________
D		**Direction :** Positive Negative None **Strength :** Strong Weak **r :** ________

Interpreting Regression Lines & r-Values

Each description on the left is written about the linear regression findings on the right. Fill in the blanks using the information in the line of best fit and the r-value.

1	For every additional Marvel Universe movie released each year, the average person is predicted to consume ________ [amount] ________ [more / fewer] pounds of sugar! This correlation is ____________ [strong, moderate, weak, practically non-existent].	$f(x) = -3.19x + 12$ $r = -0.05$
2	Shoe size and height are ____________ [strongly, moderately, weakly, not], ____________ [positively / negatively] correlated. If person A is one size bigger than person B, we predict that they will be roughly ________ [amount] inches taller than person B as well.	$f(x) = 1.65x + 52$ $r = 0.89$
3	There is ____________ [a strong, a moderate, a weak, no] relationship found between the number of Uber drivers in a city and the number of babies born each year.	$f(x) = -15.3x + 1150$ $r = 0.01$
4	The correlation between weeks-of-school-missed and SAT score is ____________ [strong, moderate, weak, practically non-existent] and ____________ [positive / negative]. For every week a student misses, we predict a more than a ________ [amount] point ________ [gain / drop] in their SAT score.	$f(x) = -5.35x - 16$ $r = -0.65$
5	There is a ____________ [strong, moderate, weak, practically non-existent], ____________ [positive / negative] correlation between the number of streaming video services someone has, and how much they weigh. For each service, we expect them to be roughly ________ [amount] pounds heavier.	$f(x) = 1.6x + 140$ $r = 0.12$

Regression Analysis in the Animals Dataset

1) I performed a linear regression on a sample of cats from the shelter and found a
dataset or subset

moderate (r=0.566), positive correlation between
weak/strong/moderate (R=…), positive/negative

age of the cats (in years) and number of weeks to adoption.
[x-axis] [y-axis]

I would predict that a 1 year increase in age is associated with a
[x-axis units] [x-axis]

0.23 week increase in adoption time.
[slope, y-units] [increase/decrease] [y-axis]

---

2) I performed a linear regression on a sample of ______ and found a
dataset or subset

______ correlation between
weak/strong/moderate (R=…), positive/negative

______ and ______.
[x-axis] [y-axis]

I would predict that a 1 ______ increase in ______ is associated with a
[x-axis units] [x-axis]

______ ______ in ______.
[slope, y-units] [increase/decrease] [y-axis]

---

3) I performed a linear regression on a sample of ______ and found a
dataset or subset

______ correlation between
weak/strong/moderate (R=…), positive/negative

______ and ______.
[x-axis] [y-axis]

I would predict that a 1 ______ increase in ______ is associated with a
[x-axis units] [x-axis]

______ ______ in ______.
[slope, y-units] [increase/decrease] [y-axis]

Regression Analysis in Your Dataset

My Dataset is ______________________________.

1) I performed a linear regression on ______________________________ and found
dataset or subset
______________________________ correlation between
a weak/strong/moderate (R=…), positive/negative
______________________________ and ______________________________.
[x-axis] [y-axis]
I would predict that a 1 ______________ increase in ______________ is associated with a
[x-axis units] [x-axis]
______________ ______________ in ______________.
[slope, y-units] [increase/decrease] [y-axis]

---

2) I performed a linear regression on ______________________________ and found
dataset or subset
______________________________ correlation between
a weak/strong/moderate (R=…), positive/negative
______________________________ and ______________________________.
[x-axis] [y-axis]
I would predict that a 1 ______________ increase in ______________ is associated with a
[x-axis units] [x-axis]
______________ ______________ in ______________.
[slope, y-units] [increase/decrease] [y-axis]

---

3) I performed a linear regression on ______________________________ and found
dataset or subset
______________________________ correlation between
a weak/strong/moderate (R=…), positive/negative
______________________________ and ______________________________.
[x-axis] [y-axis]
I would predict that a 1 ______________ increase in ______________ is associated with a
[x-axis units] [x-axis]
______________ ______________ in ______________.
[slope, y-units] [increase/decrease] [y-axis]

What’s on your mind?

Case Study: Ethics, Privacy, and Bias

My Case Study is ______________________________

1) Read the case study you were assigned, and write your summary here.

2) Is this a good thing or a bad thing? Why?

3) What are the arguments on *each* side?

Data Science used for this purpose is good because…

Data Science used for this purpose is bad because…

Threats to Validity

Threats to Validity can undermine a conclusion, even if the analysis was done correctly.

Some examples of threats are:

- **Selection bias** - identifying the favorite food of the rabbits won't tell us anything reliable about what all the animals eat.
- **Study bias** - If someone is supposed to assess how much cat food is eaten each day on average, but they only measure how much cat food is put in the bowls (instead of how much is actually consumed), they'll end up with an over-estimate.
- **Poor choice of summary** - Suppose a different shelter that had 10 animals recorded adoption times (in weeks) as 1, 1, 1, 7, 7, 8, 8, 9, 9, 10. Using the mode (1) to report what's typical would make it seem like the animals were adopted much quicker than they really were, since 7 out of 10 animals took at least 7 weeks to be adopted.
- **Confounding variables** - Some shelter workers might prefer cats, and steer people towards cats as a result. This would make it appear that "cats are more popular with people", when the real variable dominating the sample is what *workers at the shelter* prefer.

Identifying Threats to Validity

Some volunteers from the animal shelter surveyed a group of pet owners at a local dog park. They found that almost all of the owners were there with their dogs. From this survey, they concluded that dogs are the most popular pet in the state.

What are some possible threats to the validity of this conclusion?

The animal shelter noticed a large increase in pet adoptions between Christmas and Valentine's Day. They conclude that at the current rate, there will be a huge demand for pets this spring.

What are some possible threats to the validity of this conclusion?

Identifying Threats to Validity

The animal shelter wanted to find out what kind of food to buy for their animals. They took a random sample of two animals and the food they eat, and they found that spider and rabbit food was by far the most popular cuisine!

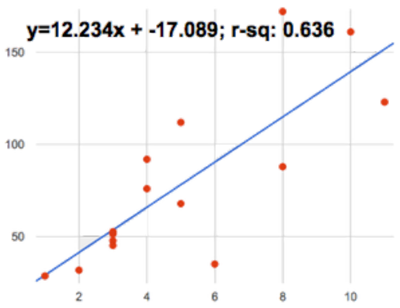
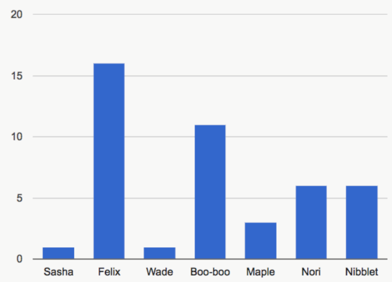
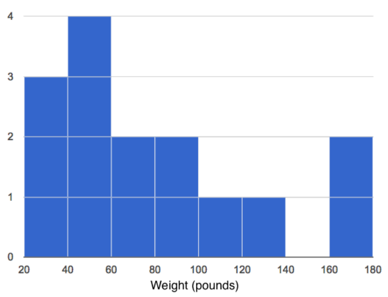
Explain why sampling just two animals can result in unreliable conclusions about what kind of food is needed.

A volunteer opens the shelter in the morning and walks all the dogs. At mid-day, another volunteer feeds all the dogs and walks them again. In the evening, a third volunteer walks the dogs a final time and closes the shelter. The volunteers report that the dogs are much friendlier and more active at mid-day, so the shelter staff assume the second volunteer must be better with animals than the others.

What are some possible threats to the validity of this conclusion?

Fake News

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

	Data	Claim	What's Wrong
1	The average player on a basketball team is 6'1".	*"Most of the players are taller than 6'."*	
2	Linear regression found a positive correlation (r=0.42) between people's height and salary.	*"Taller people are more qualified for their jobs."*	
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		*"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		*"According to this histogram, most animals weigh between 40 and 60 pounds."*	
6	Linear regression found a negative correlation (r= -0.91) between the number of hairs on a person's head and their likelihood of owning a wig.	*"Owning wigs causes people to go bald."*	

Lies, Darned Lies, and Statistics

1) Using real data and displays from your dataset, come up with a misleading claim.

2) Trade papers with someone and figure out why their claims are wrong!

Data	Claim	Why it's wrong
a		
b		
c		
d		

What's on your mind?

Design Recipe

Directions:

Contract and Purpose Statement

Every contract has three parts...

______________ :: ____ -> ______________

function name | domain | range

____

what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

______________ (______________) **is** ______________________________

function name | input(s) | what the function produces

______________ (______________) **is** ______________________________

function name | input(s) | what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun ______________ (______________________):

function name | variable(s)

____

what the function does with those variable(s)

end

Directions:

Contract and Purpose Statement

Every contract has three parts...

______________ :: ____ -> ______________

function name | domain | range

____

what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

______________ (______________) **is** ______________________________

function name | input(s) | what the function produces

______________ (______________) **is** ______________________________

function name | input(s) | what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun ______________ (______________________):

function name | variable(s)

____

what the function does with those variable(s)

end

Design Recipe

Directions:

Contract and Purpose Statement

Every contract has three parts...

\# ____________ :: ____________ -> ____________
function name domain range

\# ____________
what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

____________ (____________) **is** ____________
function name input(s) what the function produces

____________ (____________) **is** ____________
function name input(s) what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun ____________ (____________):
function name variable(s)

____
what the function does with those variable(s)

end

Directions:

Contract and Purpose Statement

Every contract has three parts...

\# ____________ :: ____________ -> ____________
function name domain range

\# ____________
what does the function do?

Examples

Write some examples, then circle and label what changes...

examples:

____________ (____________) **is** ____________
function name input(s) what the function produces

____________ (____________) **is** ____________
function name input(s) what the function produces

end

Definition

Write the definition, giving variable names to all your input values...

fun ____________ (____________):
function name variable(s)

____
what the function does with those variable(s)

end

Design Recipe

Directions:

Contract and Purpose Statement

Every contract has three parts...

__________ :: __________ -> __________

function name | domain | range

____

what does the function do?

Examples

Write some examples, then circle and label what changes...

```
examples:
  ________ ( ________ ) is ________
```

function name | input(s) | what the function produces

```
  ________ ( ________ ) is ________
```

function name | input(s) | what the function produces

```
end
```

Definition

Write the definition, giving variable names to all your input values...

```
fun ________ ( ________ ):
```

function name | variable(s)

____

what the function does with those variable(s)

```
end
```

Directions:

Contract and Purpose Statement

Every contract has three parts...

__________ :: __________ -> __________

function name | domain | range

____

what does the function do?

Examples

Write some examples, then circle and label what changes...

```
examples:
  ________ ( ________ ) is ________
```

function name | input(s) | what the function produces

```
  ________ ( ________ ) is ________
```

function name | input(s) | what the function produces

```
end
```

Definition

Write the definition, giving variable names to all your input values...

```
fun ________ ( ________ ):
```

function name | variable(s)

____

what the function does with those variable(s)

```
end
```

Contracts

Contracts tell us how to use a function. For example: `num-min :: (a :: Number, b :: Number) -> Number` tells us that the name of the function is `num-min`, it takes two inputs (both Numbers), and it evaluates to a `Number`. From the contract, we know `num-min(4, 6)` will evaluate to a `Number`. **Use the blank line under each contract for notes or sample code for that function!**

Name		Domain		Range
num-sqr	::	Number	->	Number
num-sqr(9)				
num-sqrt	::	Number	->	Number
num-sqrt(25)				
triangle	::	Number, String, String	->	Image
triangle(80, "solid", "darkgreen")				
circle	::		->	Image
star	::		->	
square	::		->	
rectangle	::		->	
text	::		->	
ellipse	::		->	

Contracts

Contracts tell us how to use a function. For example: `num-min :: (a :: Number, b :: Number) -> Number` tells us that the name of the function is `num-min`, it takes two inputs (both Numbers), and it evaluates to a `Number`. From the contract, we know `num-min(4, 6)` will evaluate to a `Number`. **Use the blank line under each contract for notes or sample code for that function!**

Name		Domain		Range
; `regular-polygon`	::		->	
# `rhombus`	::		->	
# `right-triangle`	::		->	
# `isosceles-triangle`	::		->	
# `radial-star`	::		->	
# `star-polygon`	::		->	
`overlay`	::	`Image, Image`	->	`Image`
overlay(star(30, "solid", "gold"),circle(30, "solid", "blue"))				
beside	::	Image, Image	->	Image
beside(star(50, "solid", "orange"),circle(50, "solid", "green"))				
above	::	Image, Image	->	Image
above(triangle(30, "solid", "red"),square(30, "solid", "blue"))				

Contracts

Contracts tell us how to use a function. For example: `num-min :: (a :: Number, b :: Number) -> Number` tells us that the name of the function is `num-min`, it takes two inputs (both Numbers), and it evaluates to a `Number`. From the contract, we know `num-min(4, 6)` will evaluate to a `Number`. **Use the blank line under each contract for notes or sample code for that function!**

Name		Domain		Range
`put-image`	::	`Image, Number, Number, Image`	->	`Image`
`put-image(star(30, "solid", "red"), 50, 150, rectangle(300, 200, "outline", "black"))`				
`rotate`	::	`Number, Image`	->	`Image`
`rotate(35, rectangle(30, 80, "solid", "purple"))`				
`scale`	::	`Number, Image`	->	`Image`
`scale( 0.8, triangle(30, "solid", "red"))`				
`string-repeat`	::	`String, Number`	->	`String`
`string-repeat("cheetah ", 5)`				
`string-contains`	::	`String, String`	->	`Boolean`
`string-contains("rockstar", "star")`				
`num-min`	::	`Number, Number`	->	`Number`
`num-min(80, 20)`				
`num-max`	::	`Number, Number`	->	`Number`
`num-max(80, 20)`				
`count`	::	`Table, String`	->	`Table`
`count(animals-table, "species")`				
`mean`	::	`Table, String`	->	`Number`
`mean(animals-table, "age")`				

Contracts

Contracts tell us how to use a function. For example: `num-min :: (a :: Number, b :: Number) -> Number` tells us that the name of the function is `num-min`, it takes two inputs (both Numbers), and it evaluates to a `Number`. From the contract, we know `num-min(4, 6)` will evaluate to a `Number`. **Use the blank line under each contract for notes or sample code for that function!**

Name		Domain		Range
median	::	Table, String	->	Number
median(animals-table, "age")				
modes	::	Table, String	->	List<Number>
modes(animals-table, "age")				
bar-chart	::	Table, String	->	Image
bar-chart(animals-table, "legs")				
pie-chart	::	Table, String	->	Image
pie-chart(animals-table, "species")				
histogram	::	(t :: Table, column :: String, bin-width :: Number)	->	Image
histogram(animals-table, "age", 2)				
box-plot	::	Table, String	->	Image
box-plot(animals-table, "age")				
modified-box-plot	::	Table, String)	->	Image
modified-box-plot(animals-table, "age")				
scatter-plot	::	(t :: Table, labels :: String, xs :: String, ys :: String)	->	Image
scatter-plot(animals-table, "species", "pounds", "weeks")				
image-scatter-plot	::	(t :: Table, xs :: String, ys :: String, f :: (Row -> Image))	->	Image
image-scatter-plot(animals-table, "pounds", "weeks", animal-img)				

Contracts

Contracts tell us how to use a function. For example: `num-min :: (a :: Number, b :: Number) -> Number` tells us that the name of the function is `num-min`, it takes two inputs (both Numbers), and it evaluates to a `Number`. From the contract, we know `num-min(4, 6)` will evaluate to a `Number`. **Use the blank line under each contract for notes or sample code for that function!**

Name		Domain		Range
r-value	::	(t :: Table, xs :: String, ys :: String)	->	Number
r-value(animals-table,"pounds", "weeks")				
lr-plot	::	(t :: Table, labels :: String, xs :: String, ys :: String)	->	Image
lr-plot(animals-table, "species", "pounds", "weeks")				
random-rows	::	(t :: Table, num-rows :: Number)	->	Table
random-rows(animals-table, 5)				
<Table>.row-n	::	Number	->	Row
animals-table.row-n(5)				
<Table>.order-by	::	(col :: String, increasing :: Boolean)	->	Table
animals-table.order-by("species", true)				
<Table>.filter	::	(test :: (Row -> Boolean))	->	Table
animal-table.filter(is-cat)				
<Table>.build-column	::	(col :: String, builder :: (Row -> Any))	->	Table
animals-table.build-column("sticker", label)				
bar-chart-summarized	::	(t :: Table, labels :: String, values :: String)	->	Image
bar-chart-summarized(animals-table, "species", "pounds")				
pie-chart-summarized	::	(t :: Table, labels :: String, values :: String)	->	Image
pie-chart-summarized(animals-table, "age", "pounds")				

Contracts

Contracts tell us how to use a function. For example: `num-min :: (a :: Number, b :: Number) -> Number` tells us that the name of the function is `num-min`, it takes two inputs (both Numbers), and it evaluates to a `Number`. From the contract, we know `num-min(4, 6)` will evaluate to a `Number`. **Use the blank line under each contract for notes or sample code for that function!**

Name		Domain		Range
	::		->	
	::		->	
	::		->	
	::		->	
	::		->	
	::		->	
	::		->	
	::		->	
	::		->	

9 781716 654817

These materials were developed partly through support of the National Science Foundation, (awards 1042210, 1535276, 1648684, and 1738598),