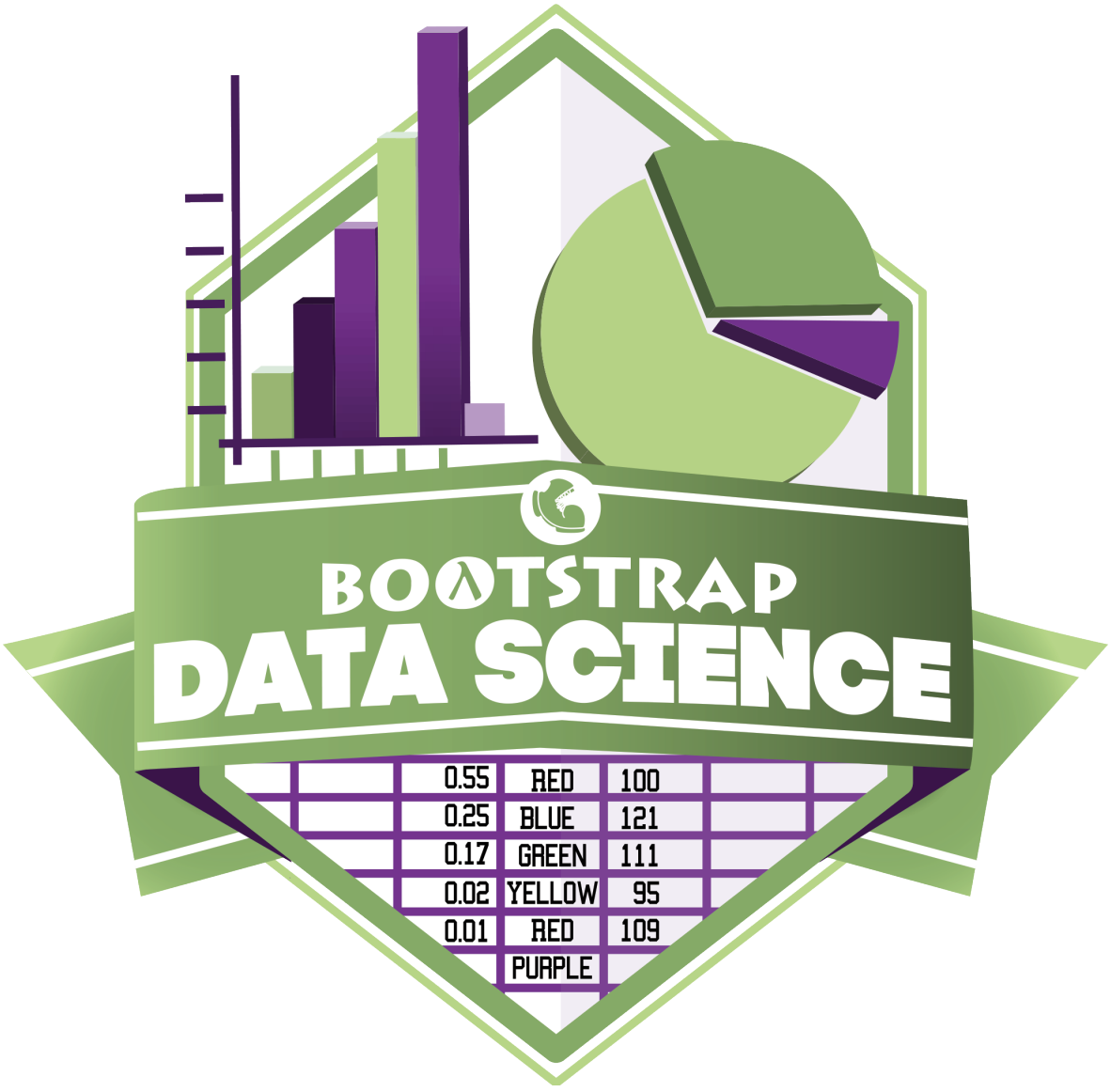


BOOTSTRAP
www.bootstrapworld.org

Workbook v1.4

Brought to you by the Bootstrap team:

- Emmanuel Schanzer
- Emma Youndtsmith
- Jennifer Poole
- Kathi Fisler
- Shriram Krishnamurthi
- Joe Politz
- Ben Lerner
- Nancy Pfenning

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

Unit 1

Intro 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 *quantity*, such as a person's height, a score on test, a measure of distance, etc. A list of quantitative data can be ordered from smallest to largest.
- **Categorical data** is data that specifies *categories*, such as eye color, country of origin, etc. Categorical data is not subject to the laws of arithmetic – for example, we cannot take the "average" of a list of colors.

Programming languages involve different *datatypes*, such as Numbers, Strings, Booleans and Images. Numbers are usually used for quantitative data, and other values are used as categorical data.

- **Operators** (like +, -, *, <, etc.) are written between values. For example: `4 + 2`
- We can use **functions** (like triangle, star, string-repeat, etc.) by writing the function name first, followed by a list of **arguments** in parentheses. For example: `star(50, "solid", "red")`
- Functions have **contracts**, which specify the *Name, Domain and Range* of each function. The Domain tells us what type of data the function consumes, and the Range tells us what it produces.

The Animals Dataset

What do you NOTICE about the animals dataset?	What do you WONDER about the animals dataset?

1. This dataset is Animals from an animal shelter, which contains 31 data rows.
2. Some of the columns are:
   1. species, which contains categorical data. Some example values from this column are: "cat", "dog", and "rabbit".
   2. _____, which contains _____ data. Some example values from this column are: _____.
   3. _____, which contains _____ data. Some example values from this column are: _____.

Numbers and Strings

Make sure you've loaded the Animals Starter File, and clicked "Run".

1. Try typing `42` into the Interactions Area and hitting "Enter". What happens?

2. Try typing in other Numbers. What happens if you try a decimal like `0.5`? A fraction like `1/3`? Try really big Numbers, and really small ones.

3. String values are always in quotes. Try typing your name (in quotes!). What happens when you hit Enter?

4. Try typing your name with the opening quote, but *without* the closing quote. What happens? Now try typing it without *any* quotes.

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

Operators

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

---

7. Type in the following expressions, one at a time: `4 + 2 + 6`, `4 + 2 * 6`, and `4 + (2 * 6)`. What do you notice? Write your answer below:

---

8. Try typing in `4 + "cat"`, and then `"dog" + "cat"`. What can you conclude from this? Write your answer below:

Booleans

Boolean expressions are yes-or-no questions, and will always evaluate to either `true` ("yes") or `false` ("no"). What will each of the expressions below evaluate to? Write down the result in the blanks provided, and type them into Pyret if you're not sure.

`3 <= 4`	_____	`"a" > "b"`	_____
`3 == 2`	_____	`"a" <> "b"`	_____
`2 <> 4`	_____	`"a" == "b"`	_____
`3 <> 3`	_____	`"a" <> "a"`	_____

Boolean Operators

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

```
(3 <= 4) and (3 == 2)
```
_____

```
("a" == "b") and (3 <> 4)
```
_____

```
(3 <= 4) or (3 == 2)
```
_____

```
("a" == "b") or (3 <> 4)
```
_____

1. How many different Number values are there in Pyret? _____
2. How many different String values are there in Pyret? _____
3. How many different Boolean values are there in Pyret? _____

Unit 2

Questions and Definitions

Answering Questions from Data can take many forms. Here are a few types of questions, each requiring a different kind of analysis:

- **Lookup Questions** can be answered just by finding the right row and column a table. (e.g. – "How old is Toggle?")

- **Compute Questions** can be answered by computing over a single row or column. (e.g. – "What is the heaviest animal at the shelter?")

- **Relate Questions** require looking for trends across multiple rows or columns. (e.g. – "Do cats tend to be adopted sooner than dogs?")

Methods are special functions that are attached to pieces of data. We use them to manipulate Tables. They are different from functions in several ways:

- Their names can't be used alone: they can only be used as part of data, separated by a dot. (For example, `shapes.row-n(2)`)
- Their contracts are different: they include the type of the data as part of their names. (eg, `<table>.row-n :: (index :: Number) → Row`)
- They have a "secret" argument, which is the data they are attached to.
- In this course, the methods we'll be using are `row-n, order-by, filter,` and `build-column`.

We can **define our own functions**, using a technique called the **Design Recipe**.

- We use the Design Recipe to help us define functions **and think through problems clearly.**

- The first step is to write a **Contract** and **Purpose Statement** for the function, which specify the Name, Domain and Range of the function and give a summary of what it does.

- The second step is to **write at least two examples**, which show how the function should work for specific inputs. These examples help us see patterns, and we express those patterns by **circling and labeling** what changes.

- The final step is to **define the function**, which generalizes our examples.

Lookup Questions

The table below represents four pets at an animal shelter:

animals-table

name	gender	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?"	`animals-table.row-n(3)`
"Which is the last row in the table?	`animals-table.row-n(2)["name"]`
"What is Fritz's gender?"	`animals-table.row-n(1)["gender"]`
"What's the third animal's name?"	`animals-table.row-n(3)["age"]`
"How much does Nori weigh?"	`animals-table.row-n(3)["pounds"]`
"How old is Maple?"	`animals-table.row-n(0)`
"What is Toggle's gender?"	`animals-table.row-n(2)["pounds"]`
"What is the first row in the table?"	`animals-table.row-n(0)["gender"]`

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

animals-table.row-n(3)["name"]	"Maple"
_____	"male"
_____	4
_____	48
_____	"Nori"

More Practice with Lookups

Consider the table below, and the four value definitions that follow:

shapes-table

name	corners	is-round
"triangle"	3	false
"square"	4	false
"rectangle"	4	false
"circle"	0	true

```
shapeA = shapes-table.row-n(0)
shapeB = shapes-table.row-n(1)
shapeC = shapes-table.row-n(2)
shapeD = shapes-table.row-n(3)
```

---

1. ***Match*** each Pyret expression (left) to the description of what it looks up (right).

`shapeD`	Evaluates to `4`
`shapeA`	Evaluates to the last row in the table
`shapeB["corners"]`	Evaluates to `"square"`
`shapeC["is-round"]`	Evaluates to `true`
`shapeB["name"]`	Evaluates to `false`
`shapeA["corners"]`	Evaluates to `3`
`shapeD["name"] == "circle"`	Evaluates to the first row in the table

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

a. _____	`"rectangle"`
b. _____	`"triangle"`
c. _____	`4`
d. _____	`0`
e. _____	`true`

The Design Recipe

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

Define a function called `is-fixed`, which <u>looks up</u> whether or not an animal is fixed.

```
# ___is-fixed___ :: ___(animal :: Row)___ → ___Boolean___
      name                domain                range
# Consumes an animal, and looks up the value in the fixed column
examples:

    ____________(_________) is ____________________________

    ____________(_________) is ____________________________
end

fun ____________ (________): ____________________________

end
```

Define a function called `gender`, which consumes a Row of the animals table and <u>looks up</u> the gender of that animal.

```
# ________________::________________________ → ________________
      name                domain                range
# ____________________________________________________________
examples:

    ____________(_________) is ____________________________

    ____________(_________) is ____________________________
end

fun ____________ (________): ____________________________

end
```

The Design Recipe

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

Define a function called `is-cat`, which consumes a Row of the animals table and <u>computes</u> whether the animal is a cat.

```
# ___is-cat___ :: ___(animal :: Row)___ → ___Boolean___
    name              domain                range
# Consumes an animal, looks up the species column, and computes if species is "cat"
examples:

    ___is-cat___(___animalA___) is ___________________________

    ___________(_________) is ___________________________
end

fun ___________ (________): ___________________________

end
```

Define a function called `is-young`, which consumes a Row of the animals table and <u>computes</u> whether it is less than four years old.

```
# ___________ :: ___________________ → ___________
    name              domain                range
# ___________________________________________________
examples:

    ___________(_________) is ___________________________

    ___________(_________) is ___________________________
end

fun ___________ (________): ___________________________

end
```

Unit 3

Exploring Datasets

Computer Scientists may take **samples** that are subsets of a data set. 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!

My Dataset

What do you NOTICE?	What do you WONDER?	Question Type
		Lookup Compute Relate
		Lookup Compute Relate
		Lookup Compute Relate
		Lookup Compute Relate
		Lookup Compute Relate
		Lookup Compute Relate

1. This dataset is _____, which contains _____ data rows.
2. Some of the columns are:
    1. _____, which contains _____ data, and is of type _____. Some example values from this column are: _____.
    2. _____, which contains _____ data, and is of type _____. Some example values from this column are: _____.
    3. _____, which contains _____ data, and is of type _____. Some example values from this column are: _____.

Samples from My Dataset

What are some subsets you can create from this dataset? For a given row *r*, what code will identify if that row is in the subset? An example from the Animals Dataset is written in the first row, to get you started.

Subset		A single row *r* is in the subset if…
Kittens		`(r["age"] < 2) and (r["species"] == "cat")`

Design Recipes – Filtering Rows

Use your work from the previous page to help you write filter functions for your dataset, which you can use to define subsets.

Define a function called _____ , which consumes a Row of the _____ table and _____

_____

```
# ________________::________________________ → ________________
      name                  domain                    range
# ____________________________________________________________
examples:

    ____________(________) is ________________________________

    ____________(________) is ________________________________
end

fun ____________ (________) :  ________________________________

end
```

Define a function called _____ , which consumes a Row of the _____ table and _____

_____

```
# ________________::________________________ → ________________
      name                  domain                    range
# ____________________________________________________________
examples:

    ____________(________) is ________________________________

    ____________(________) is ________________________________
end
```

Design Recipes – Filtering Rows

Write your own word problems below, and solve them using the Design Recipe.

Define a function called _____ , which consumes a Row of the _____ table and _____

_____

```
# ________________::________________________ → ________________
      name                  domain                    range
# ____________________________________________________________
examples:

    ____________(_________) is ____________________________

    ____________(_________) is ____________________________
end

fun ____________ (________) :  ____________________________

end
```

Define a function called _____ , which consumes a Row of the _____ table and _____

_____

```
# ________________::________________________ → ________________
      name                  domain                    range
# ____________________________________________________________
examples:

    ____________(_________) is ____________________________

    ____________(_________) is ____________________________
end
```

Unit 4

Visualizing the "Shape" of Data

Bar charts show the number of rows belonging to a given category. The more rows in each category, the longer 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 number of orders for 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 that fall within a particular "bin", the taller the bar.

- *Histograms provide a visual representation of the 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.

Design Recipe

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

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

```
# ______________::_____________________ → ______________
      name                domain                 range
# ____________________________________________________________
examples:

    ____________(_________) is ____________________________

    ____________(_________) is ____________________________
end

fun ____________ (________):  ____________________________

end
```

Define a function called `nametag`, which consumes a Row of the animals table and <u>computes</u> an image that shows the animal's name in big, red letters.

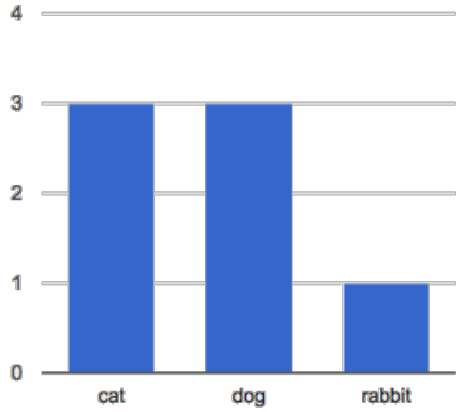
```
# ___nametag___::___(animal :: Row)___ → ___Image___
      name                domain                 range
# Consumes an animal, and produces an image of their name in big, red letters
examples:

    ___nametag___(___animalB___) is ____________________________

    ____________(_________) is ____________________________
end

fun ____________ (________):  ____________________________

end
```

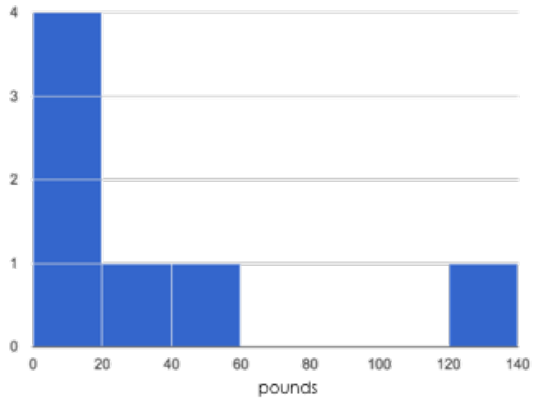
Summarizing Columns

name	species	age	pounds
"Sasha"	"cat"	1	6.5
"Boo-boo"	"dog"	11	123
"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 based on this table. What is similar about them? What is different?





Similarities	Differences

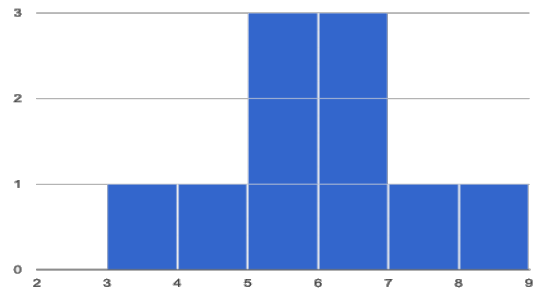
Reading Histograms

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 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.**

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

1

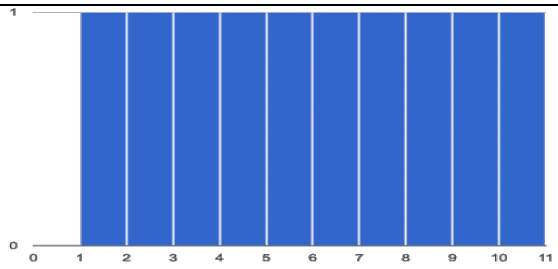
A



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

2

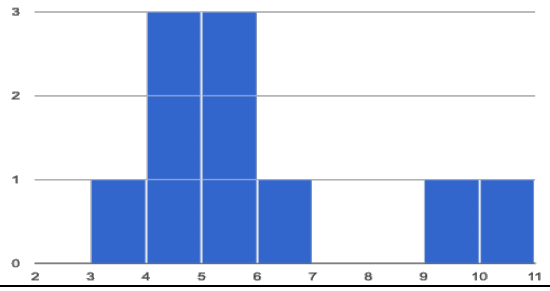
B



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

3

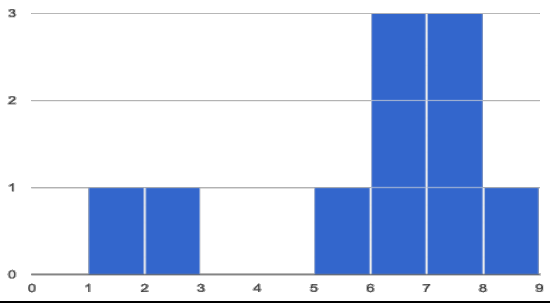
C



Students either really liked or really disliked the video.

4

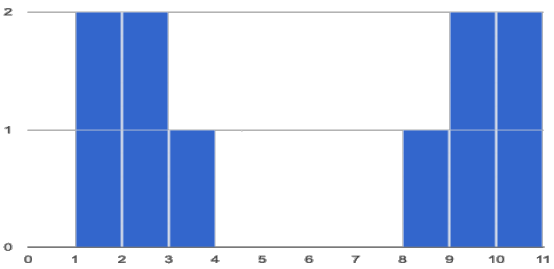
D



Reactions to the video were all over the place: high ratings and low ratings and in-between ratings were all equally likely.

5

E

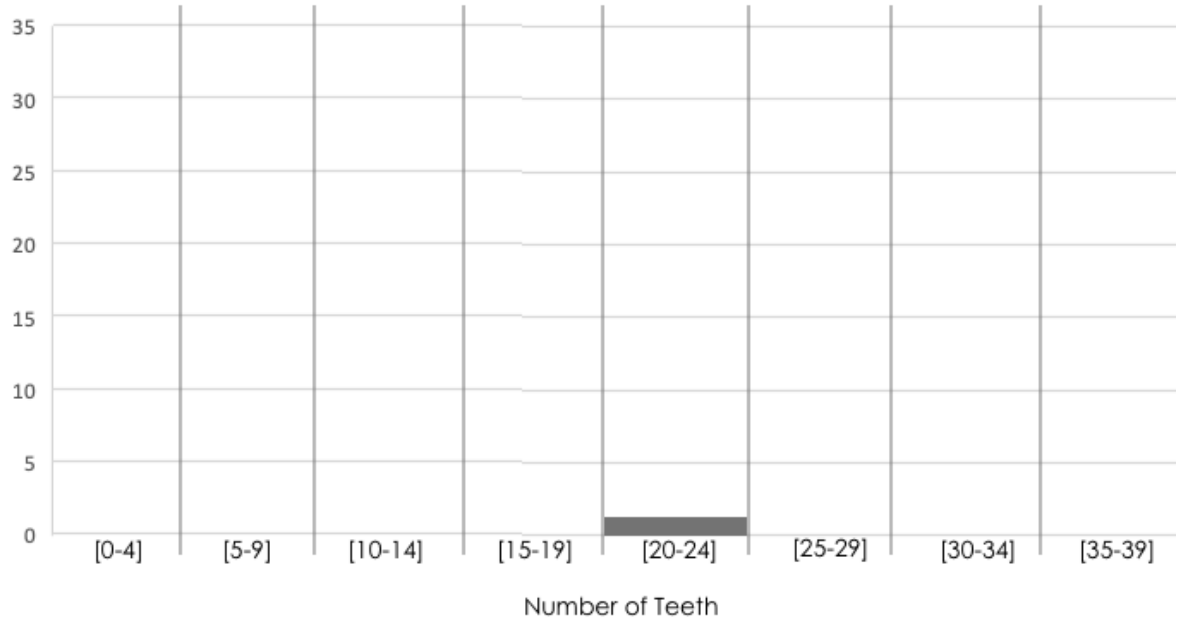


Making Histograms

Suppose we have a data set for number of teeth in a group of 50 adults:

Number of teeth	Count
0	1
22	1
26	1
27	1
28	4
29	3
30	3
31	3
32	33

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.



The Shape of the `Animals` Dataset

Describe two of the histograms you made from your dataset.

1) I made a histogram, showing the distribution of _____pounds_____ for
column in your dataset
_____animals at the shelter_____.
your subset (for example, "fixed dogs at the shelter")

2) I made a histogram, showing the distribution of _____ for

_____.

In the table below, describe the histograms. Are they symmetric? Do they show left skewness and/or low outliers? Right skewness and/or high outliers?

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

The Shape of My Dataset

Describe two of the histograms you made from your dataset.

3) I made a histogram, showing the distribution of _____ for
column in your dataset

_____.
your subset (for example, "fixed dogs at the shelter")

4) I made a histogram, showing the distribution of _____ for

_____.

In the table below, describe the histograms. Are they symmetric? Do they show left skewness and/or low outliers? Right skewness and/or high outliers?

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

Unit 5

Center and Spread

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

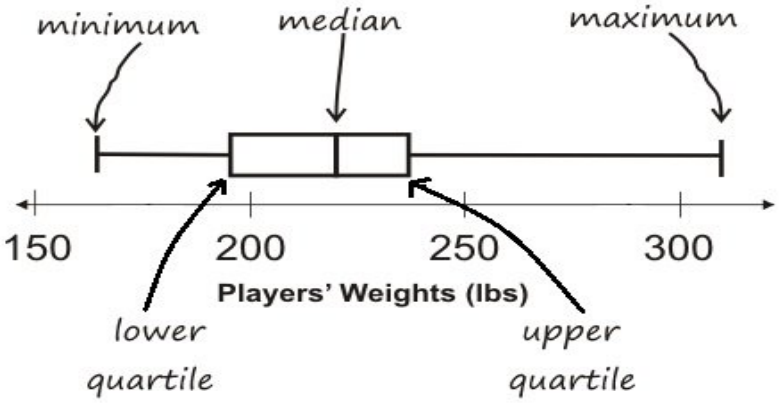
    - The **mean** of a dataset is the average of all the numbers.
    - The **median** of a dataset is a value that is smaller than half the dataset, and larger than the other half.
    - The **mode(s)** of a dataset is the value (or values) that occurs most often.

- The **shape** of a data set tells us which values are more or less common. In a *symmetric* data set, values are just as likely to occur a certain distance above or below the mean. A data set with left skewness and/or low outliers has a few values that are unusually low, pulling the mean *below* the median. Right skewness and/or high outliers means there are a few values that are unusually high, pulling the mean *above* the median.

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

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

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



Summarizing Columns in `Animals`

1) The column I choose to measure is _____pounds_____

Measures of Center

The three measures for this column are:

Mean (Average)	Median	Mode(s)

2) Since the mean is _____ than the median, this suggests that there may
[higher/lower]

be outliers or skewness due to values that are unusually _____.
[high / low]

Measures of Spread

My five-number summary is:

Minimum	Q1	Q2 (Median)	Q3	Maximum

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

From this summary and box-plot, I conclude:

Interpreting Spread

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

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

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

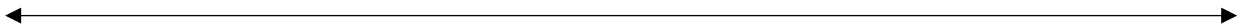
2. In the table below, compute the **measures of center** for this dataset.

Mean (Average)	Median	Mode(s)

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

Minimum	Q1	Q2 (Median)	Q3	Maximum

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



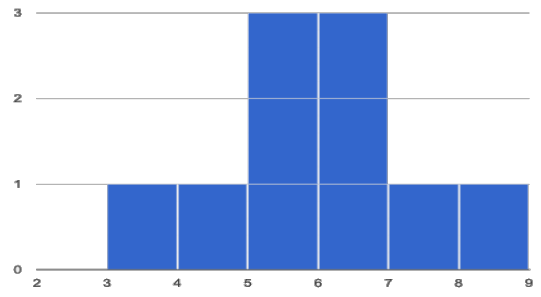
5. The following statements are *correct*…but misleading. Write down the reason why.

Statement	Why it's misleading
"They're rich! The average person makes more than $70k dollars!"	
"It's a middle-income list: the most common salary is $45k/yr!"	
"This group is really diverse, with people making as little as 12k and as much as $280k!"	

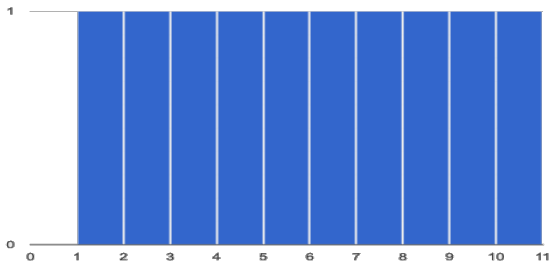
Matching Box-Plots to Histograms

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 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.**

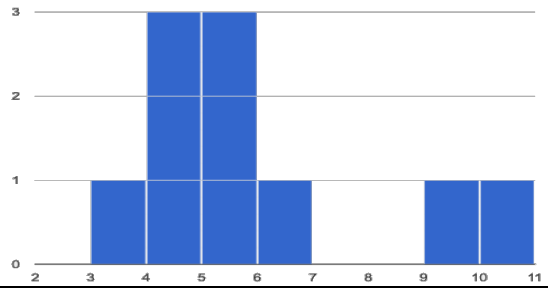
_____ **1** **A**



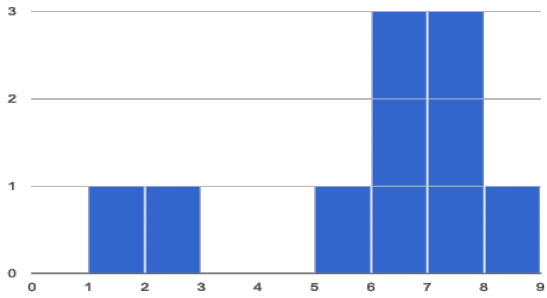
_____ **2** **B**



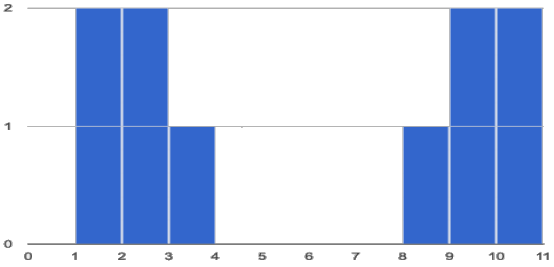
_____ **3** **C**



_____ **4** **D**



_____ **5** **E**



Summarizing a Column in My Dataset

1) The column I choose to measure is _____

Measures of Center

The three measures for this column are:

Mean (Average)	Median	Mode(s)

2) Since the mean is _____ than the median, this suggests that there may
[higher/lower]

be outliers or skewness due to values that are unusually _____.
[high / low]

Measures of Spread

My five-number summary is:

Minimum	Q1	Q2 (Median)	Q3	Maximum

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

From this summary and box-plot, I conclude:

Unit 6

Advanced Analysis

- **Method chaining** allows us to apply multiple method with less code:

  For example, we can use method-chaining to write this:

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

  Instead of using multiple definitions, like this:

```
with-labels = table.build-column("labels", nametag)
dogs = with-labels.filter(is-dog)
dogs.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.

- Data Scientists have to know whether or not they can trust their tools. Fortunately, then can *use* Data Science to **verify** that their tools do what they're supposed to!

Chaining Methods

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

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

The table **t** below represents four animals at the shelter:

name	gender	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)
```

Produces a table containing *only* Toggle and Maple

```
t.filter(is-fixed)
```

Produces a table, sorted oldest-to-youngest

```
t.build-column(“sticker”, nametag)
```

Produces a table, sorted youngest-to-oldest

```
t.filter(is-young)
```

Produces a table with an extra column, named “sticker”

```
t.order-by(“age”, false)
```

Produces a table containing Maple and Toggle, in that order

```
t.filter(is-young)
 .order-by(“pounds”, false)
```

Produces a table containing the same four animals

```
t.build-column(“label”, nametag)
 .order-by(“age”, true)
```

Won’t run: will produce an error

```
t.order-by(“gendr”, false)
```

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(animal): animal[“gender”] == “female”   end
fun kilograms(animal): animal[“pounds”] / 2.2         end
fun is-heavy(animal):  animal["kilograms"] > 25       end
```

The table **t** below represents four animals at the shelter:

name	gender	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)
```

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)
```

```
t.build-column(“kilos”, kilograms)
   .filter(is-heavy)
```

Produces a table containing only Fritz.

```
t.filter(is-heavy)
   .build-column(“kilos”, kilograms)
```

Won’t run: will produce an error

```
t.build-column(“kilos”, kilograms)
   .filter(is-heavy)
   .order-by(“gender”, true)
```

Produces a table containing only Fritz, with two extra columns.

```
t.build-column(“female”, is-female)
 .build-column(“kilos”, kilograms)
 .filter(is-heavy)
```

Produces a table containing Maple and Fritz

```
t.order-by(“name”, false)
 .build-column(“kilos”, kilograms)
 .filter(is-female)
```

Produces a table containing Maple, Nori and Toggle (in that order)

"Trust, but verify…"

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

```
# fixed-cats :: (animals :: 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?**

HINT:

- 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 that has multiple cats *not* in order 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. *(Remember: the first data row is always index zero!)*

Name

"Trust, but verify…"

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

```
# old-dogs-nametags:: (animals :: Table) → Table
# consumes a table of animals, and produces a table containing only
# dogs 10 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. *(Remember: the first data row is always index zero!)*

Name

Unit 7

Visualizing Relationships

- **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 is possible that a linear relationship exists between those two columns. A number called a **correlation** can be used to summarize this relationship.

- The correlation is **positive** if the point cloud slopes up as it goes farther to the right. 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. If they are loosely scattered, it is a **weak** correlation.

- 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 each of the points 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."

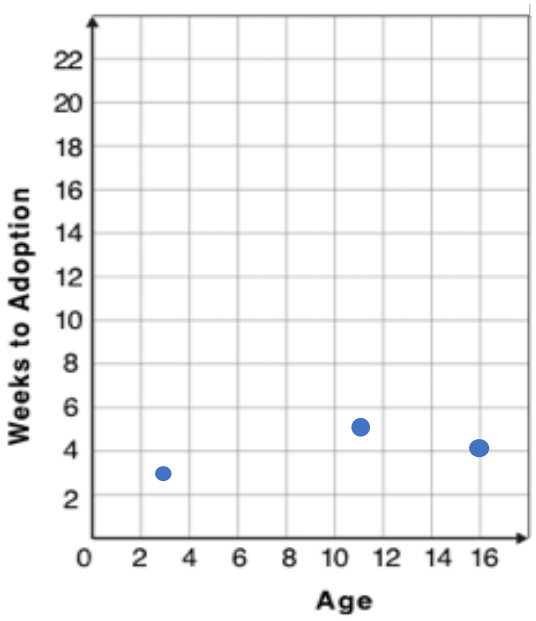
Do you agree? If so, why?

I hypothesize…

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

Creating a Scatter Plot

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



1. **For each row in the Sample Table on the left, add a point to the scatter plot on the right**. The first 3 rows have been completed for you. Use the values from the `age` column for the x-axis, and values from the `weeks` column for the y-axis.

2. Do you see a pattern? Do the points seem to shift up or down as age increases? **Draw a line on the scatter plot to show this pattern**.

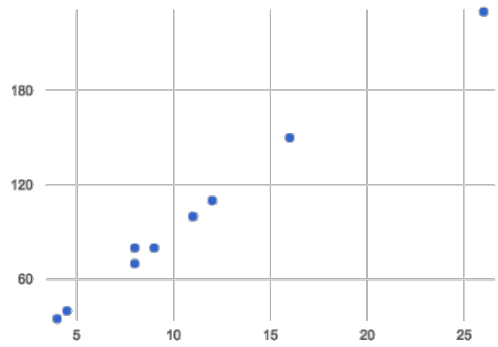
3. Does the line slope upwards or downwards? _____

4. Are the points clustered around the line? Loosely scattered? _____

Drawing Predictors

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

A

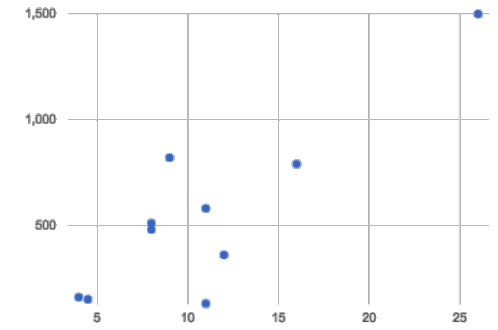


fat (g) v. calories-from-fat in common menu items

Direction: Positive Negative None

Strength: Strong Weak

B

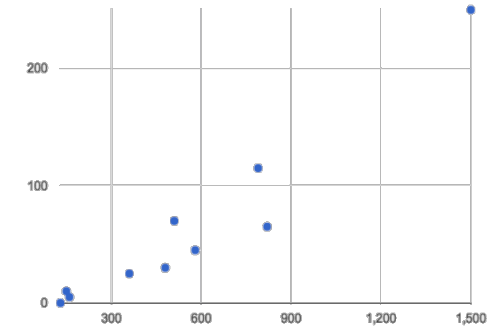


fat (g) v. sodium (g) in common menu items

Direction: Positive Negative None

Strength: Strong Weak

C

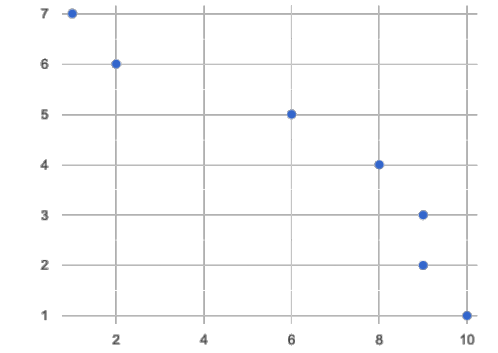


sodium (g) v. cholesterol (mg) in common menu items

Direction: Positive Negative None

Strength: Strong Weak

D



sandcastles left on the beach v. number of days

Direction: Positive Negative None

Strength: Strong Weak

Correlations in My Dataset

1) There *may* be a correlation between _____ and
column

_____. I think it is a _____, _____
column strong / weak positive / negative

correlation, because _____

_____. It might be stronger if I looked

at _____.
a subset or extension of my data

---

2) There *may* be a correlation between _____ and
column

_____. I think it is a _____, _____
column strong / weak positive / negative

correlation, because _____

_____. It might be stronger if I looked

at _____.
a subset or extension of my data

---

3) There *may* be a correlation between _____ and
column

_____. I think it is a _____, _____
column strong / weak positive / negative

correlation, because _____

_____. It might be stronger if I looked

at _____.
a subset or extension of my data

Unit 8

Computing Relationships

- **Linear Regression** is a way of computing the **line of best fit**, which minimizes the sum of squared vertical distances of all scatter plot points from 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 y-variable to increase or decrease, for each unit that the x-variable increases
    - **R** is the name of the correlation statistic, which is also computed by linear regression. The r-value will always fall between -1 and +1. The sign tells us whether the correlation is positive or negative, and distance from 0 tells us the strength of the correlation (-1 or +1 is really strong, 0 means no correlation)

- **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, air conditioners do not *cause* hot days!

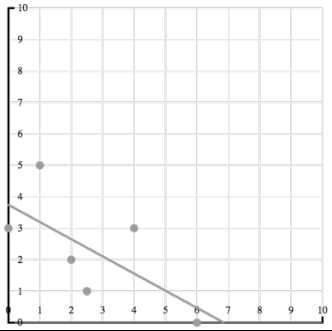
- **Sample size matters!** The number of data values is also relevant. We'd be more convinced of a positive relationship in general between cat age and time to adoption if a correlation of +0.57 were based on 50 cats instead of 5.

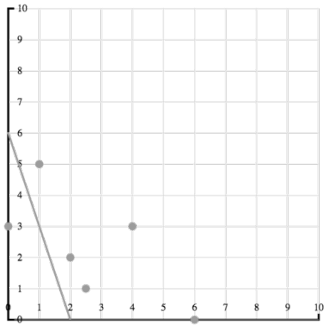
Grading Predictors

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

1. Circle the plot with the line that fits better
2. Give the circled plot your best guess for what r is closest to: -1, -0.5, +0.5, or +1.

A

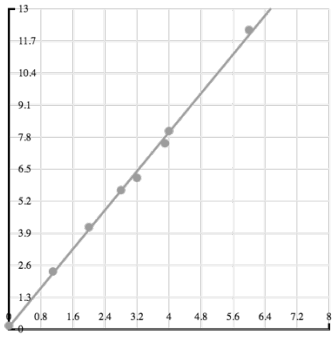


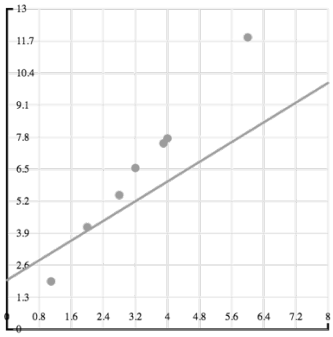


r =

-1 -0.5 +0.5 +1

B

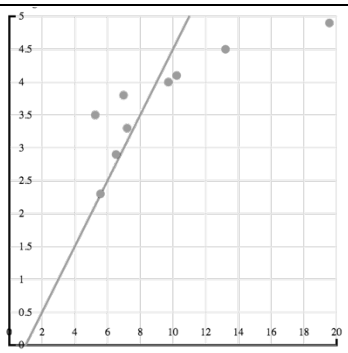


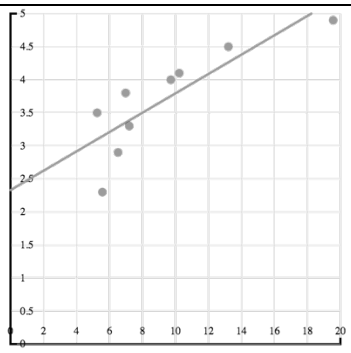


r =

-1 -0.5 +0.5 +1

C

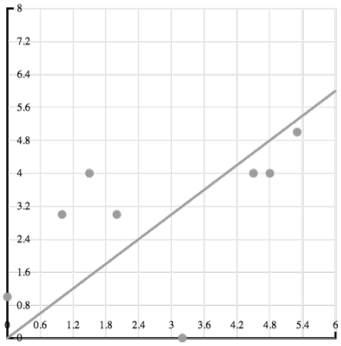


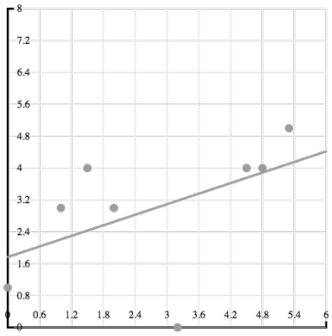


r =

-1 -0.5 +0.5 +1

D





r =

-1 -0.5 +0.5 +1

Reading Regression Lines & *r*-Values

Match the summary description (left) with the line of best fit and r-value (right).

The correlation between weeks-of-school-missed and SAT score is moderate and negative. For every week a student misses, we predict a more than a 5-point drop in their SAT score.	**1**	**A**	$y = -3.19x + 12$ $r = -0.05$
There is a weak, positive correlation between the number of streaming video services someone has, and how much they weigh. For each service, we expect them to be roughly 1.6 pounds heavier.	**2**	**B**	$y = 2.5x - 2.8$ $r = 0.89$
Foot size and height are strongly, positively correlated. If person A is one size bigger than person B, we predict that they will be roughly two and a half inches taller than person B as well.	**3**	**C**	$y = 0.012x + 7.8$ $r = 0.01$
For every additional Marvel Universe movie released each year, the average person is predicted to consume more than three pounds less sugar! However, this correlation is extremely weak.	**4**	**D**	$y = -5.35x - 16$ $r = -0.65$
There is virtually no relationship found between the number of Uber drivers in a city and the number of babies born each year.	**5**	**E**	$y = 1.6x + 160$ $r = 0.12$

Regression Analysis in the `animals` Dataset

I performed a linear regression on cats at the shelter (dataset or subset), and found a moderate (r=0.566), positive (a weak/strong/moderate (R=__), positive/negative) correlation between age of the cats (in weeks) [x-axis] and number of weeks to adoption [y-axis]. I would predict that a 1 year [x-axis units] increase in age [x-axis] is associated with a 0.23 week [slope, y-units] increase [increase/decrease] in adoption time [y-axis].

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

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

Regression Analysis in My Dataset

I performed a linear regression on _____, and
dataset or subset

found _____ correlation between
a weak/strong/moderate (R=__), positive/negative

_____ and _____. I would predict that
[x-axis] [y-axis]

a 1 _____ increase in _____ is associated with a _____
[x-axis units] [x-axis] [slope, y-units]

_____ in _____.
[increase/decrease] [y-axis]

I performed a linear regression on _____, and
dataset or subset

found _____ correlation between
a weak/strong/moderate (R=__), positive/negative

_____ and _____. I would predict that
[x-axis] [y-axis]

a 1 _____ increase in _____ is associated with a _____
[x-axis units] [x-axis] [slope, y-units]

_____ in _____.
[increase/decrease] [y-axis]

I performed a linear regression on _____, and
dataset or subset

found _____ correlation between
a weak/strong/moderate (R=__), positive/negative

_____ and _____. I would predict that
[x-axis] [y-axis]

a 1 _____ increase in _____ is associated with a _____
[x-axis units] [x-axis] [slope, y-units]

_____ in _____.
[increase/decrease] [y-axis]

Unit 9

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.

- **Sample size** – averaging the age of only three animals won't tell us anything reliable about the age of animals at the shelter!

- **Sample error** – surveying dogs when they are puppies won't tell us anything reliable about overall dog behavior, since their behavior changes as they age.

- **Confounding variables** – shelter workers might steer people towards newer animals, because they've become attached to the animals that have been there for a while, making it *appear* that "staying at the shelter longer" means "less likely to be adopted".

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, and from this survey, they concluded that dogs are the most popular pet in the region.

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?

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!

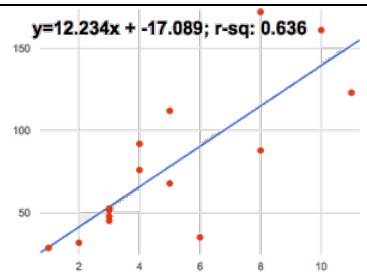
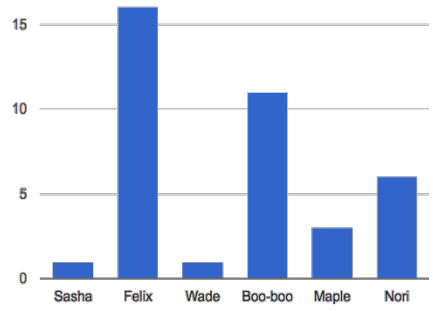
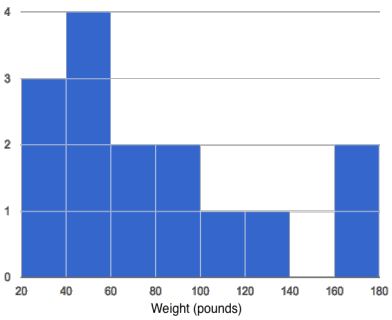
What are some possible threats to the validity of this conclusion?

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

Lies, Darned Lies, and Statistics…

- Using real data and displays from your dataset, come up with a misleading claim.
- Trade papers with someone and figure out why their claims are wrong!

	Data	**Claim**	**Why it's *wrong***
1			
2			
3			
4			

Blank Recipes and References

Design Recipes

```
# ______________::_____________________ → ______________
      name                 domain                 range
# ____________________________________________________________
examples:

    ____________(________) is ______________________________

    ____________(________) is ______________________________
end

fun ____________ (_______):  ______________________________

end
```

```
# ______________::_____________________ → ______________
      name                 domain                 range
# ____________________________________________________________
examples:

    ____________(________) is ______________________________

    ____________(________) is ______________________________
end

fun ____________ (_______):  ______________________________

end
```

Design Recipes

```
# ______________::_______________________ → ______________
      name                 domain                 range
# ____________________________________________________________
examples:

    ____________(________) is ______________________________

    ____________(________) is ______________________________
end

fun ____________ (_______):  ______________________________

end
```

```
# ______________::_______________________ → ______________
      name                 domain                 range
# ____________________________________________________________
examples:

    ____________(________) is ______________________________

    ____________(________) is ______________________________
end

fun ____________ (_______):  ______________________________

end
```

Design Recipes

```
# ______________::_______________________ → ______________
      name                 domain                 range
# ____________________________________________________________
examples:

    ____________(________) is ______________________________

    ____________(________) is ______________________________
end

fun ____________ (_______):  ______________________________

end
```

```
# ______________::_______________________ → ______________
      name                 domain                 range
# ____________________________________________________________
examples:

    ____________(________) is ______________________________

    ____________(________) is ______________________________
end

fun ____________ (_______):  ______________________________

end
```

Contracts

Contracts tell us how to use a function. For example: `num-sqr :: (n :: Number) → Number` tells us that the name of the function is **`num-sqr`**, that it takes one input (a **`Number`**), and that it evaluates to a number. From the contract, we know **`num-sqr(4)`** will evaluate to a **`Number`**.

Name	Domain		Range
`triangle`	`:: (side-length :: Number, style :: String, color :: String)`	→	*Image*
`circle`	`:: (radius :: Number, style :: String, color :: String)`	→	*Image*
`star`	`:: (radius :: Number, style :: String, color :: String)`	→	*Image*
`rectangle`	`:: (width :: Num, height :: Num, style :: Str, color :: Str)`	→	*Image*
`ellipse`	`:: (width :: Num, height :: Num, style :: Str, color :: Str)`	→	*Image*
`square`	`:: (size-length :: Number, style :: String, color :: String)`	→	*Image*
`text`	`:: (str :: String, size :: Number, color :: String)`	→	*Image*
`overlay`	`:: (img1 :: Image, img2 :: Image)`	→	*Image*
`rotate`	`:: (degree :: Number, img :: Image)`	→	*Image*
`scale`	`:: (factor :: Number, img :: Image)`	→	*Image*
`string-repeat`	`:: (text :: String, repeat :: Number)`	→	*String*
`string-contains`	`:: (text :: String, search-for :: String)`	→	*Boolean*
`num-sqr`	`:: (n :: Number)`	→	*Number*
`num-sqrt`	`:: (n :: Number)`	→	*Number*
`num-min`	`:: (a :: Number, b:: Number)`	→	*Number*
`num-max`	`:: (a :: Number, b:: Number)`	→	*Number*

Contracts

Contracts tell us how to use methods. For example: `<Table>.filter :: (test :: (Row→Boolean)) → Table` tells us that the name of the function is **`.filter`** and that it is a **`Table`** method. The domain says it has one input (a function that consumes **`Rows`** and produces **`Booleans`**), and that the method evaluates to a Table.

Name	Domain		Range
count	:: (t :: *Table*, col :: *String*)	→	*Table*
random-rows	:: (t :: *Table*, num-rows :: *Number*)	→	*Table*
<Table>.row-n	:: (n :: *Number*)	→	*Row*
<Table>.order-by	:: (col :: *String*, *increasing* :: *Boolean*)	→	*Table*
<Table>.filter	:: (test :: *(Row → Boolean)*)	→	*Table*
<Table>.build-column	:: (col :: *String*, *builder* :: *(Row → Value)*)	→	*Table*
mean	:: (t :: *Table*, col :: *String*)	→	*Number*
median	:: (t :: *Table*, col :: *String*)	→	*Number*
modes	:: (t :: *Table*, col :: *String*)	→	*List<Number>*
bar-chart	:: (t :: *Table*, col :: *String*)	→	*Image*
pie-chart	:: (t :: *Table*, col :: *String*)	→	*Image*
bar-chart-raw	:: (t :: *Table*, labels :: *String*, values :: *String*)	→	*Image*
pie-chart-raw	:: (t :: *Table*, labels :: *String*, values :: *String*)	→	*Image*
box-plot	:: (t :: *Table*, col:: *String*)	→	*Image*
histogram	:: (t :: *Table*, values :: *String*, bin-width :: *Number*)	→	*Image*
scatter-plot	:: (t :: *Table*, labels :: *String*, xs :: *String*, ys :: *String*)	→	*Image*
lr-plot	:: (t :: *Table*, labels :: *String*, xs :: *String*, ys :: *String*)	→	*Image*