Data Science

thomas mcandrew

January 22, 2022

Contents

List of Figures ix

List of Tables xi

I Probability 1

1 Sets, the sample space, and probability 3
thomas mcandrew
- 1.1 Introduction . . . 3
- 1.2 Probability lingo . . . 4
  - 1.2.1 Sets . . . 4
- 1.3 Applying set theory to probability . . . 6
  - 1.3.1 Foundation . . . 6
  - 1.3.2 The principle of equally likely outcomes—one way to assign probabilities . . . 7
  - 1.3.3 A frequentist approach to assigning probabilities . . . 8
  - 1.3.4 Products, Conditionals, Baye's Theorem, and Repeated Experiments . . . 9
    - 1.3.4.1 Product Sets . . . 9
    - 1.3.4.2 Conditional probabilities . . . 10
    - 1.3.4.3 Repeated experiments . . . 12
  - 1.3.5 Product, tuples, and relational data bases . . . 13
- 1.4 Exercises . . . 13
- 1.5 Glossary . . . 14

2 Laboratory 01 15
thomas mcandrew, david braun
- 2.1 Jupyter Notebooks . . . 15
  - 2.1.1 File . . . 16
    - 2.1.1.1 A new notebook . . . 16
    - 2.1.1.2 The notebook . . . 16
    - 2.1.1.3 Save your work . . . 16
    - 2.1.1.4 Export for submission . . . 16
  - 2.1.2 Kernel . . . 17
- 2.2 Programming and R . . . 18
- 2.3 Arithmetic . . . 18
- 2.4 Vectors . . . 19

- 2.4.1 Assignment . . . 19
  - 2.4.1.1 c() . . . 19
  - 2.4.1.2 assign . . . 19
  - 2.4.1.3 equals . . . 19
- 2.5 Print . . . 20
- 2.6 Combining vectors . . . 20
- 2.7 Indexing and access . . . 21
  - 2.7.1 Numeric indexing . . . 21
  - 2.7.2 Logical vectors and logical indexing . . . 22
    - 2.7.2.1 True and False . . . 22
    - 2.7.2.2 Logical comparisons . . . 22
    - 2.7.2.3 Logic . . . 22
    - 2.7.2.4 AND, OR, and NOT . . . 23
    - 2.7.2.5 Equivalence of TRUE and FALSE to 1 and 0 . . . 23
  - 2.7.3 Two functions that are useul for operating on vectors . . . 23
- 2.8 Assignment 01 . . . 24
  - 2.8.1 The data . . . 24
  - 2.8.2 Please complete the following . . . 24

II Statistics **27**

Bibliography **29**

List of Figures

2.1 kernelselect.png 16
2.2 Caption 17

List of Tables

1.1 Frequencies collected about our vaccination experiment . . . 10
1.2 Caption 12
1.3 Probabilities of the cooccurance of an intervention and the rise or fall of an infectious agent, and the probability of each of these four events assuming they are independent. 12

2.1 Caption 23

Part I

Probability

1

Sets, the sample space, and probability

thomas mcandrew

Lehigh University

CONTENTS

1.1 Introduction 3
1.2 Probability lingo 3
  1.2.1 Sets 4
1.3 Applying set theory to probability 6
  1.3.1 Foundation 6
  1.3.2 The principle of equally likely outcomes—one way to assign probabilities 7
  1.3.3 A frequentist approach to assigning probabilities 8
  1.3.4 Products, Conditionals, Baye's Theorem, and Repeated Experiments 8
    1.3.4.1 Product Sets 9
    1.3.4.2 Conditional probabilities 9
    1.3.4.3 Repeated experiments 12
  1.3.5 Product, tuples, and relational data bases 13
1.4 Exercises 13
1.5 Glossary 14

1.1 Introduction

When we decide to study a topic, we often have in mind a population, a protocol to observe members of our population, a hypothesis that posits some aspects of our observed members. We collect data, we compile results. At the close of our study we want to know how the results ("what happened") contribute to our hypotheses ("what we thought may happen") and, importantly, the chances that if we reproduced the same study that the results would lead us towards the same conclusions.

For example, suppose we decide to study the impact of percutaneous intervention (PCI) versus optimal medical therapy (OMT) to treat patients how have had a myocardial infarction, often called a "heart attack". Patients are enrolled in a study and randomly assigned PCI or OMT. We hypothesize that PCI will result in a prolonged life from when the intervention took place onward. After our study we find patients who underwent PCI live on average 100 days longer than those who underwent OMT. However, it feels unsatisfactory to have a single number describing how effective PCI was at prolong life. We may want to know the chances PCI prolongs life 80 days, 30 days, or maybe zero days, compared to OMT.

We want to understand how likely we are to see specific events. The formal method of assigning chances to a set of events is called probability theory.

1.2 Probability lingo

1.2.1 Sets

We define a **set** as a collection of items, sometimes called elements. A set is typically given a capital letter (for example A) and the elements are included inside curly braces.

Here, we define three sets

$$A = \{a, b, c\} \tag{1.1}$$

$$C = \{-1, 1, 3.14\} \tag{1.2}$$

$$Z = \{\text{orange}, 1, \text{puppy dog}\} \tag{1.3}$$

The elements inside a set are unordered and do not necessarily need to be numbers. A set can contain any object. The set A could have been defined as $\{a, b, c\}$ or $\{c, b, a\}$. We use the symbol $\in$ to express that an element "is a member of" a set. For example, $a \in A$ can be read "the element a is a member of the set A" or "the element a is in the set A". We can also communicate when an element is not a member of a set with the symbol $\notin$. For example, $c \notin C$ is read "the element c is not a member of (not in) te set C".

Two sets are **equal** is they contain the same elements. In other words, if for an element x, $x \in A$ implies that $x \in B$ **and** if for an element y, $y \in B$ implies that $y \in A$ then the set A and set B are equal. If a set A and set B are equal we write $A = B$. If two sets do contain the same elements than they are **unequal** and we write $A \neq B$.

Lets look at an example.

$$
\begin{align}
A &= \{a, b, c\} \tag{1.4}\\
B &= \{b, a, c\} \tag{1.5}\\
C &= \{b, c\} \tag{1.6}
\end{align}
$$

Above, $A = B$ but $A \neq C$ and $B \neq C$.

A set A is a **subset** of a second set B if all of the elements in A are also elements in B. We can say that A is a subset of B if every element in x implies that element is in B, or $x \in A$ implies $x \in B$. We write $A \subset B$ to denote that A is a subset of B. To denote that a set A is **not a subset** of B, we write $A \not\subset B$. In the example above, the C is a subset of A $C \subset A$, but A is not a subset of C, or $A \not\subset C$.

We can build new sets by operating on two or more existing sets.

Set **intersection** takes as input two or more sets, A,B, and returns a set that contains all the elements that are in both A and B. We use the symbol "cap" to denote set intersection $A \cap B$ and often say "A intersect B".

For the above sets A and C, their intersection is

$$A \cap C = \{c, b\} \tag{1.7}$$

because the element c belongs to both set A and set C and because the element b belongs to A and C.

We can intersect more than two sets.

Let $A = \{a, b, 10, 12\}$, $Z = \{b, 10, -1\}$, $U = \{a, b, c, d, 10\}$. Then

$$A \cap Z = \{b, 10\} \tag{1.8}$$

and

$$A \cap Z \cap U = \{b, 10\}. \tag{1.9}$$

If A is a subset of B, $A \subset B$, then the only elements both sets have in common are those in A. The intersection of A and B is then

$$
\begin{aligned}
\text{If } A &\subset B \text{ then}\\
A \cap B &= A.
\end{aligned}
\tag{1.10}
$$

Set **union** takes as input two or more sets and output a set that contains all the elements that belong to first set or the second set or the third set, and so on. We use the "cup" symbol to denote set union. As an example, consider the sets A, Z, and U to be the same as they are above. Then A **union** Z is

$$A \cup Z = \{a, b, 10, 12, -1\} \tag{1.11}$$

This is because each of the above elements belongs to at least one of the sets A and Z. As another example,

$$A \cup Z \cup U = \{a, b, c, 10, 12, -1\} \tag{1.12}$$

Intersection and union are the most common set operations. Before we define the set operation **complement**, we need to discuss two special sets.

The **universal set** is the set of all elements. We denote the universal set with a $\mathcal{G}$. The set which contains no elements is called the **empty set**, and we denote the empty set as $\emptyset$.

If a set A and set B have no elements in common, then $A \cap B = \emptyset$. We often say that the set A and B are **disjoint**. For example,

$$\begin{aligned} A &= \{1, 3, 5\} \\ Q &= \{2, 4, 6\} \\ A \cap Q &= \emptyset \end{aligned} \tag{1.13}$$

Because the intersection between A and Q is empty, these sets are disjoint.

Now for our final set operation—complement. The set complement takes as input a single set, A, and outputs a set with elements that are members of $\mathcal{G}$ but are not members of A. We denote set complement in one of two ways: A^c or A'.

Set operations example

Let's look at an example of how these sets and set operations work. Define the universal set, $\mathcal{G}$, to be the set of all positive integers (i.e. $1, 2, 3, 4, \cdots$). Lets define the sets $A_1 = \{1\}, A_2 = \{1, 2\}, A_3 = \{1, 2, 3\}, \cdots, A_n = \{1, 2, 3, 4 \cdots, n\}$. We use a subscript and a number to "index" our sets. An index is an easy way to keep track of sets (and many mathematical objects).

$$\begin{aligned} A_1 \cap A_2 &= \{1\} \\ A_1 \cap A_5 &= \{1\} \\ A_3 \cap A_8 &= \{3\} \end{aligned} \tag{1.14}$$

Note that $A_3 \cap A_8$ is not the value 3, but the set {3}. The examples above suggest a pattern for any set A_i and A_j where $i \leq j$:

$$A_i \cap A_j = \{i\} \tag{1.15}$$

Lets look at set union in this example

$$\begin{aligned} A_1 \cup A_2 &= \{1, 2\} \\ A_1 \cup A_5 &= \{1, 2, 3, 4, 5\} \\ A_3 \cup A_8 &= \{1, 2, 3, 4, 5, 6, 7, 8\} \end{aligned} \tag{1.16}$$

and we see the following pattern for $i \leq j$

$$A_i \cup A_j = \{1, 2, 3, 4 \cdots, j\} = A_j \tag{1.17}$$

Because we defined a universal set, we can look at set complement

$$\begin{aligned} A_1^c &= \{2, 3, 4, 5, \cdots\} \\ A_5^c &= \{5, 6, 7, 8, \cdots\} \\ A_8^c &= \{8, 9, 10, 11, 12, \cdots\} \end{aligned} \tag{1.18}$$

1.3 Applying set theory to probability

1.3.1 Foundation

The ideas about sets and set operations are the foundations for how we think about probability, about experiments, and hypotheses. We only need to recast the above set ideas to results from an experiment.

We use $\mathcal{G}$ to define the set of all possible outcomes from an experiment and call this set the **sample space**. The term "experiment" has a broad meaning. An experiment can mean everything from a randomized controlled trial to an observational study. Here an **experiment** is the process that generates observations.

An **outcome** is defined as an element of the sample space. An outcome is a single observation from an experiment, and we define an **event** as a set of outcomes. Most often we use o_i to denote an outcome and E_i to denote an event.

The **probability** of an event E is a mapping from the set E to a number between, or equal to, the values 0 and 1. We use $P(E)$ to denote the probability of event E. We require that the probabilities assigned to individual sets consisting of a single element in E add up to the probability of E. Lets suppose there are n outcomes in the event E. Then

$$\begin{aligned} E &= \{o_1, o_2, o_3, \cdots, o_n\} \\ P(E) &= P(\{o_1\}) + P(\{o_2\}) \cdots P(\{o_n\}) \end{aligned} \tag{1.19}$$

We further require $P(\mathcal{G}) = 1$. In other words, the probability that something happens is certain. Note that we do not need to describe how we assign probabilities to events, we only describe what values we expect those numbers to be.

Lets further detail relationships between probabilities of sets that we would expect.

1. If $A \subset B$ then $P(A) \leq P(B)$.
2. $P(A \cup B) \leq P(A) + P(B)$ (why?)
3. $P(A \cup B) = P(A) + P(B) - P(A \cap B)$
4. If A and B are disjoint then $P(A \cup B) = P(A) + P(B)$

The following three axioms (knowledge we assume true without proof) are called **Kolmorogov's axioms** ()

1. For any event E, $P(E) \geq 0$
2. $P(\mathcal{G}) = 1$
3. If E_1 and E_2 are disjoint then $P(E_1 \cup E_2) = P(E_1) + P(E_2)$

Example: When we want to compute the probability that an outcome fall in any one of the events, say E_1, E_2, E_3, knowing that these events have no outcomes in common, that they are disjoint, makes the computation easy. An intuitive way to think about events that are disjoint is if they cannot all occur at the same time. Suppose we wish to study the prevalence of stroke among patients who are older than sixty-five. Strokes can be categorized into transient ischemia attacks (TIA), ischemic stroke (IS), and hemorrhagic stroke (HS). Further lets assume the following probabilities for each event: $P(\{\text{TIA}\}) = 0.15$, $P(\{\text{IS}\}) = 0.75$, $P(\{\text{HS}\}) = 0.10$, and that these events cannot occur simultaneously. Then the probability of the event $E = \{TIA, HS\}$ can be broken into the union of two disjoint events $E = \{TIA\} \cup \{HS\}$ and we can use what we know about these disjoint events to compute the probability of E

$$\begin{aligned} P(E) &= P(\{\text{TIA}, \text{HS}\}) && (1.20)\\ &= P(\{\text{TIA}\} \cup \{\text{HS}\}) && (1.21)\\ &= P(\{\text{TIA}\}) + P(\{\text{HS}\}) \quad \text{disjoint} && (1.22)\\ &= 0.15 + 0.75 = 0.90 && (1.23) \end{aligned}$$

1.3.2 The principle of equally likely outcomes—one way to assign probabilities

There are many different ways to assign probabilities to events. In this text, we will only consider using the principle of equally likely outcomes to assign probabilities. The **principle of equally likely outcomes** (PELO) states that the probability we assign to every event which contains a single outcome, $E_i = \{o_i\}$, in a sample space $\mathcal{G}$ is equal.

We can use this principle to assign probabilities first to every individual outcome and then to arbitrary events. Assume the PELO is true, then

$$\begin{aligned} o_1, o_1, o_1, \cdots, o_n \in \mathcal{G} && (1.24)\\ E_i = \{o_i\} && (1.25)\\ E_1 \cup E_2 \cup E_3 \cup \cdots \cup E_n = \mathcal{G} \ \text{(why?)} && (1.26)\\ P(E_1 \cup E_2 \cup \cdots \cup E_n) = P(\mathcal{G}) = 1 && (1.27)\\ P(E_1 \cup E_2 \cup \cdots \cup E_n) = P(E_1) + P(E_2 \cup \cdots \cup E_n) \ \text{(disjoint)} && (1.28)\\ P(E_1) + P(E_2) + \cdots P(E_n) = 1 && (1.29)\\ n \cdot p = 1 \ \ (p \text{ is the constant prob. we are after}) && (1.30)\\ p = 1/n && (1.31) \end{aligned}$$

The PELO tells us that the probability of an event with a single outcome is equal to one divided by the total number of outcomes in the sample space. The PELO also tells us that for a sample space with n elements the probability of an event $E = \{o_1, o_{20}, o_4\}$ that contains three outcomes is $P(E) = \frac{3}{20}$.

An example when, and when not, the PELO applies.

PELO only works when we expect each outcome in our sample space to be equally likely.

Example 1 Suppose our experiment is a coin toss and we will observe whether or not the coin lands heads up (H) or tails up (T). If we assume the coin has not been altered in any way, then PELO applies. Our sample space is $\mathcal{G} = H, T$. Because there are two outcomes in our sample space, $P(\{H\}) = 1/2$ and $P(\{T\}) = 1/2$.

Example 2 Suppose for our experiment we observe for one year a single patient who was admitted to the hospital because of an influenza infection and we plan to observe if that patient returns to the hospital because of a second infection. Our sample space contains one outcome R if the patient is re-admitted to the hospital within one year and a second outcome if they are no re-admitted (N). Our sample space is $\mathcal{G} = R, N$. If we applied the PELO to this problem we would have to assume the probability this patient returns, or does not return, are equal. Intuitively,it feels unreasonable to assume these two events have equal probabilities.

We need an additional way to assign probabilities.

1.3.3 A frequentist approach to assigning probabilities

An empirical approach to assign probabilities to an event A with sample space $\mathcal{G}$ is to do the following:

1. Generate $1, 2, 3, 4, \cdots, N$ outcomes from $\mathcal{G}$
2. Define a variable $N(A)$ and set that variable to zero.
3. If the 1st outcome is a member of A than add one $N(A)$, otherwise move on
4. If the 2nd outcome is a member of A than add one to $N(A)$

   $\vdots$
5. If the Nth outcome is a member of A than add one to $N(A)$

The above algorithm defined a variable $N(A)$ that counts the number of outcomes that belong to A out of the N generated outcomes. We can assign to A the probability

$$P(A) = \frac{N(A)}{N} \tag{1.32}$$

that is, the number of times we observed an outcome that belonged to A divided by the number of outcomes we observed.

As an example, suppose we want to understand the probability that patients with non-medically treated type II diabetes transition to needing treatment within one year of their diagnosis. Our outcomes are $\mathcal{G}$ = {"need treatment" and "no treatment"}. To assign a probability to the event $E = \{\text{need treatment}\}$ using the frequentist approach we decide to request anonymized patient records from a set of hospitals, asking that each hospital only send those patients who had an original diagnosis of type II diabetes without medication and who have had a physician visit one year more later. After our data request, we received 5,000 patient records and find that 1,585 of these patients were asked to start medication for their type II diabetes within one year. A frequentist approach would assign

$$P(E) = 1,585/5,000 = 0.317 \tag{1.33}$$

and also assign to the event $F = \{\text{no treament}\}$

$$P(F) = 0.683 \;\; \text{(why?)} \tag{1.34}$$

.

1.3.4 Products, Conditionals, Baye's Theorem, and Repeated Experiments

1.3.4.1 Product Sets

Suppose a novel vaccine was developed and we are asked to compare two probabilities: the probability a vaccinated patient is infected before 30 days after they receive their vaccine and the probability a patient who has not received a vaccine is infected before 30 days since they have enrolled in the experiment. To estimate these probabilities, we enroll 200 volunteer patients. We will assign 100 patients to be given the treatment and the remaining 100 patients will be observed without treatment.

Because of our experimental design, $P(\{\text{vaccinated}\}) = 1/2$ and $P(\{\text{not vaccinated}\}) = 1/2$ with a sample space $\mathcal{G}_{\text{assignment}}$, containing two outcomes $\{\text{vaccinated}, \text{not vaccinated}\}$. But our main interest is in the probability of infection within 30 days of enrolling in the study, and so our main interest is in the a related sample space $\mathcal{G}_{\text{infection}} = \{\text{infection}, \text{no infection}\}$. We need a way to combine these sample spaces together so that we can estimate the probability of a patient that is vaccinated and that is infected.

Up until this point we have learned how to assign probabilities to events in one sample space. Lets look at how we may assign events to a combination of, and later a sequence of, sample space**s**.

Products of sets: First, we'll need to talk about the product of two sets. Let

$A = \{a, b, c\}$ and $B = \{1, 2, 3\}$. Then the Cartesian product $C = A \times B$ is a set where each element in A is paired with each element in B.

$$C = \{(a,1),(a,2),(a,3),(b,1),(b,2),(b,3),(c,1)(c,2)(c,3)\} \tag{1.35}$$

We use the notation $(,)$, called a "tuple", to denote pairs. A **tuple** is an outcome belonging to a sample space built from the Cartesian product of many individual sample spaces. Tuples are ordered. The tuple $(a,1) \neq (1,a)$. The above is called a Cartesian product because you could imagine creating a grid where the horizontal axis has gridlines at "a", "b", "c" and the vertical axis has grid lines at "1", "2", and "3". Where the gridlines intersect are the tuples that belong to C.

We can apply the Cartesian product to combine samples spaces. Define $\mathcal{G}_{\text{experiment}}$ as the Cartesian product between $\mathcal{G}_{\text{assignment}}$ and $\mathcal{G}_{\text{infection}}$, or

$$\mathcal{G}_{\text{experiment}} = \mathcal{G}_{\text{assignment}} \times \mathcal{G}_{\text{infection}} \tag{1.36}$$

This new sample space has the following outcomes:

$$\begin{aligned} \mathcal{G}_{\text{experiment}} = \{&(\text{vaccinated}, \text{infection}), (\text{vaccinated}, \text{not infected}) \\ &, (\text{not vaccinated}, \text{infection}), (\text{not vaccinated}, \text{not infected})\} \end{aligned} \tag{1.37}$$

Our new sample space has 4 outcomes, two outcomes from the assignment sample space are paired with two outcomes from the infection sample space (2 outcomes × 2 outcomes equals 4 outcomes in the new space).

With this new space in hand, we can assign probabilities to the outcomes "vaccinated and infected" and the outcome "not vaccinated and infected". You may hear events like the above called **compound events**, or **joint events**.

Let's use the frequentist approach to estimate the probabilities of these four events. We enroll all 200 patients over a 6 months period and then follow each patient for 30 days. At 7 days, 14 days, 21 days, and at the end of there 30 day period they meet with a physician to relay information about how they feel that could indicate they had an infection. We collect the following data:

Vaccinated	Infected	Frequency	Estimated Prob.
Yes	Yes	20	20/200 = 0.1
Yes	No	80	80/200 = 0.4
No	Yes	40	40/200 = 0.2
No	No	60	60/200 = 0.3

From our experimental evidence we would assign a probability of $P(\{\text{vacc}, \text{infected}\}) = 0.10$ to those who were vaccinated and infected and a probability of $P(\{\text{no vacc}, \text{infected}\}) = 0.20$ to those who were not vaccinated and were infected. But this was not what were were asked to compute.

We were asked to compute the probability that someone is infected after they received (given that they already had) a vaccine, and the probability of infection given a patient was not vaccinated. Probabilities of one event, given we have observed another event are called **conditional probabilities**.

1.3.4.2 Conditional probabilities

Assume a sample space $\mathcal{G}$. We define the conditional probability of event A given event B as

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{1.38}$$

Example: Lets use the definition above to compute the probability of an event A given $\mathcal{G}$, the sample space (remember that the sample space is a set and so technically an event). By the definition of conditional probbility

$$P(A|\mathcal{G}) = \frac{P(A \cap \mathcal{G})}{P(\mathcal{G})} \tag{1.39}$$

We know that $P(\mathcal{G}) = 1$ (Kolmogorov axioms in 1.3.1) and so

$$P(A|\mathcal{G}) = \frac{P(A \cap \mathcal{G})}{P(\mathcal{G})} \tag{1.40}$$

$$P(A|\mathcal{G}) = \frac{P(A \cap \mathcal{G})}{1} \tag{1.41}$$

$$= P(A \cap \mathcal{G}) \tag{1.42}$$

The set A must be a subset of $\mathcal{G}$ and so by (1.10)

$$P(A|\mathcal{G}) = P(A \cap \mathcal{G}) = P(A) \tag{1.43}$$

The above example shows that we can think of the event that we condition on as a new sample space. Lets return to our example of vaccination and the incidence of infection.

To compute the probability of infection given a patient was vaccinated we need to compute

$$P(\text{inf}|\text{vacc}) = \frac{P(\text{inf} \cap \text{vacc})}{P(\text{vacc})} \tag{1.44}$$

where the event "inf" contains those outcomes in $\mathcal{G}_{\text{experiment}}$ which have "infected" in the fist position of their tuple $(\text{infected}, \cdot)$ and the event "vacc" contains those outcomes which have "vaccinated" in the second position of their tuple $(\cdot, \text{vaccinated})$. Then $P(\text{inf} \cap \text{vacc})$ are those outcomes with "infected" in the first position and "vaccinated" in the second position. We have a single outcome where this happens: {(infected, vaccinated)} and this outcome has a probability of $P(\text{inf} \cap \text{vacc}) = 0.1$.

Computing $P(\text{vacc})$ is only slightly more difficult. We can use what we learned about unions and disjoint events for help.

$$\begin{aligned} P(\text{vacc}) &= P(\{(\text{infected}, \text{vacc}), (\text{not infected}, \text{vacc})\}) \\ &= P(\{(\text{infected}, \text{vacc})\} \cup \{(\text{not infected}, \text{vacc})\}) \\ &= P(\{(\text{infected}, \text{vacc})\}) + P(\{(\text{not infected}, \text{vacc})\}) \\ &= 0.1 + 0.4 = 0.5 \end{aligned} \tag{1.45}$$

We arrive at our final result.

$$P(\text{inf}|\text{vacc}) = \frac{P(\text{inf} \cap \text{vacc})}{P(\text{vacc})} = \frac{0.1}{0.5} = 0.2 \tag{1.46}$$

and we can do the same for

$$P(\text{inf}|\text{no vacc}) = \frac{P(\text{inf} \cap \text{no vacc})}{P(\text{ no vacc})} = \frac{0.2}{0.5} = 0.4 \tag{1.47}$$

We report back to our experimental team that our estimated probability of infection from someone who received a vaccine is 0.2 and the probability of infection from someone who did not received a vaccine is 0.4, double the probability of a vaccinated individual.

Conditional probabilities give us, for free, an alternative way to compute the probability of the intersection of two sets. All we need to do is rearrange the definition of conditional probability

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{1.48}$$

$$P(A \cap B) = P(B)P(A|B) \tag{1.49}$$

Equation (1.49) is called the **multiplication rule**. Lets explore how the multiplication rule may make computing probabilities easier.

Example: We want to compute the probability that it rains and we remember to leave the house with an umbrella. We can imagine our sample space having the following four outcomes: $\mathcal{G}$ = {(rain,remember),(rain,forgot), (no rain, remember), (no rain, forgot)}. We try to remember all the times that the outcome (rain,remember) occurred in our past—but we cannot. Instead, we can certainly estimate the probability we forgot our umbrella given that it rained because it was an unpleasant experience (for some). We estimate $P(\text{forgot}|\text{rain}) = 0.95$. To compute $P(\text{forgot} \cap \text{rain})$ all we need now is the probability it rains. We take a look at the weather station reports over the past year and use the frequentest approach to estimate the probability it rains:$P(\text{rain}) = 0.25$. So then the probability it rains and we forget our umbrella is

$$\begin{aligned} P(\text{forgot} \cap \text{rain}) &= P(\text{forgot}|\text{rain})P(\text{rain}) \\ &= 0.95 \times 0.25 = 0.2375 \end{aligned} \tag{1.50}$$

The multiplication rule makes computations easier because we can think of events occurring in sequence. First it rains and second we forget our umbrella.

Conditional probabilities also allow us to express what it means for two events to be **independent**. An event A is independent from an event B if

$$P(A|B) = P(A) \tag{1.51}$$

We know the event B occurred, but this has not changed the probability that A has occurred. When two events are independent we can compute their intersection more easily.

$$\begin{aligned} &\text{Event } A \text{ and } B \text{ are independent} \\ &P(A \cap B) = P(A|B)P(B) && \text{multiplication rule} \\ &P(A \cap B) = P(A)P(B) && \text{independence} \end{aligned} \tag{1.52}$$

If two events are independent than the probability they occur together is the product of their probabilities. Statistical independence can be difficult to see.

Example We are recruited to track the evolution of an infectious agent for a team of public health officials (PHOs). To support future strategic planning the PHOs want to know the impact of intervention X on increases or decreases in the incidence of this infectious agent. The PHO team collected for each county in their state whether the intervention was enacted, and whether the incidence of case counts of this infectious agent increased or decreased 60 days after the intervention was in place data. When we look at this table, our first thought may be that when the intervention is enacted there is a high probability that the infection rate decreases (0.525), evidence that this is an important intervention for preventing the spread of this agent.

Intervention	Raise in infection	Probability
yes	yes	0.225
yes	no	0.525
no	yes	0.075
no	no	0.175

But be careful. Let's compute the probability of intervention $P(\{intervention\})$ and the probability of a raise in infection $P(\{raise\})$. To do this we should define our sample space to be clear about what outcomes we can observe from our experiment.

Our sample space is the set of four pairs

$$\begin{aligned} \mathcal{G} = \{&(\text{intervention}, \text{raise}), (\text{intervention}, \text{lower}) \\ &, (\text{no intervention}, \text{raise}), (\text{no intervention}, \text{lower})\} \end{aligned} \tag{1.53}$$

and we want to compute $P(\text{intervention}) = 0.225 + 0.525 = 0.75$ (why?) and $P(\text{raise}) = 0.225 + 0.075 = 0.30$ (why?).

One way that can test whether intervention is independent of a rise or fall in incidence, is compare the probabilities we estimated from our data (in table **??**) to the product of individual events we computed above. Lets look at an example. The probability of an intervention and rise in incidence was estimated from our data to be $P(\text{intervention}, \text{raise}) = 0.225$. Lets compare this probability to the product $P(\text{intervention}) \cdot P(\text{raise})$.

$$P(\text{intervention}) \cdot P(\text{raise}) = 0.75 \cdot 0.30 = 0.225 \tag{1.54}$$

We have a match. Our estimated probability of an intervention and a rise occurring together is equal to the probability that an intervention occurs and a rise in incidence occurs. We can do the same procedure above for the remaining three scenario.

Intervention	Raise	Probability from data	Probability assuming indep.
yes	yes	0.225	0.75 (0.30) = 0.225
yes	no	0.525	0.75 (0.70) = 0.525
no	yes	0.075	0.25 (0.30) = 0.075
no	no	0.175	0.25 (0.70) = 0.175

We see that the probabilities we collected form the data match the product of the probabilities of each individual event (Table **??**). Intervention is independent, does not change the probability, of a rise or fall in incidence of the infectious agent we are tracking. We will need to report these results to the PHO team and recommend they try alternative interventions to curb the spread of this agent.

1.3.4.3 Repeated experiments

Many natural experiments involve repeated observations. It may be of interest to observe the whether and assign a probability to "sunny" or "cloudy" weather conditions. However, many vacationers, who plan to spend 2, 3, or more days at a remote destination may want to know the probability that the next X days are sunny.

The Cartesian product gives us a natural way to express repeated experiments. If we chose to repeat an experiment N times where each experiment produced an outcome in $\mathcal{G}$, we could imagine the set of outcomes as tuples of length N where each entry in the tuple is from our sample space. In other words, a single outcome in this repeated experiment would be $(o_1, o_2, o_3, \cdots, o_N)$ and this outcome is a member of the set $\mathcal{G} \times \mathcal{G} \times \mathcal{G} \times \cdots \times \mathcal{G}$.

Back to our vacation example. If a vacationer wanted to understand the chances they had of enjoying 5 sunny days in a row, then we know that we could defined a sample space $\mathcal{G} = \{$"sunny", "not sunny"$\}$ and we also know that the vacationer is interested in outcomes in the sample space

$$(d_1, d_2, d_3, d_4, d_5) \in \mathcal{G} \times \mathcal{G} \times \mathcal{G} \times \mathcal{G} \times \mathcal{G} \tag{1.55}$$

where d_i is the outcome on day i. We can frame the problem, and clearly layout the potential outcomes, for example on outcome is ("sunny", "sunny", "not sunny", "sunny", "not sunny").

1.3.5 Product, tuples, and relational data bases

<to write>

1.4 Exercises

1.

$$\begin{aligned} &A = 1,2,3,4,5,6; \quad B = 1,3,6 \\ &C = 7 \quad D = \emptyset \end{aligned}$$

   (a) Please compute $A \cap B$
   (b) Please compute $A \cup C$
   (c) Please compute $A \cup D$
   (d) Please compute $A \cap D$
   (e) Please compute $(A \cap B) \cup (C \cup D)$

2. Let the sample space $\mathcal{G} = \{1,2,3,4,5,6,7\}$
   (a) Please compute A^c
   (b) Please compute B^c
   (c) Please compute D^c
   (d) Please compute $\mathcal{G} \cap A$
   (e) Is $A \subset \mathcal{G}$?
   (f) Is $\emptyset \subset \mathcal{G}$?

3. Let $A = 0, 1, 2$ for some sample space $\mathcal{G}$ that we know contains the more elements than 0,1,2. Further assume $P(A) = 0.2$.
   (a) Are the sets A and A^c disjoint? Why or why not.
   (b) Use Kolmogorov's axioms to show that $P(A) = 1 - P(A^c)$

4. If $A = \{1, 2, 3\}$ and $B = \{2, 3, 4\}$ and $C = \{1, 3\}$
    (a) Can $P(A) < P(B)$? Why or why not
    (b) Can $P(A) < P(C)$? Why or why not
5. Use what you know about the intersection, about subsets, and about probability to show that $P(A \cap B) \leq P(A)$. Hint: How are $A \cap B$ and A related?
6. Suppose we wish to study the reemergence of cancer among patients in remition. We collect data on 1,000 patients who are in cancer remition and follow them for 5 years. At five years we are interested in the probability of a second cancer.
    (a) Define a sample space $\mathcal{G}$ we can use to assign probabilities to a second cancer and no second cancer.
    (b) After five years of followup we find that 238 patients experienced a second cancer. Use the frequentist approach to assign probabilities to a second cancer and no second cancer.
    (c) If you collected data on 2,000 patients do you expect the probability of a second cancer to change? How do you expect the probability to be different for 2,000 patients than with 1,000 patients?

1.5 Glossary

Set:

Subset:

Set equality:

Set intersection:

Set union:

Universal set:

Empty set:

Sample space:

Experiment:

Outcome:

Event:

Probability:

Kolmorogov's Axioms:

Principle of Equally Likely Outcomes:

Product Set:

Compound event:

Conditional probability:

Multiplication Rule:

Independence:

2

Laboratory 01

thomas mcandrew, david braun

Lehigh University

CONTENTS

2.1 Jupyter Notebooks 15
  2.1.1 File 16
    2.1.1.1 A new notebook 16
    2.1.1.2 The notebook 16
    2.1.1.3 Save your work 16
    2.1.1.4 Export for submission 16
  2.1.2 Kernel 17
2.2 Programming and R 17
2.3 Arithmetic 18
2.4 Vectors 19
  2.4.1 Assignment 19
    2.4.1.1 c() 19
    2.4.1.2 assign 19
    2.4.1.3 equals 19
2.5 Print 19
2.6 Combining vectors 20
2.7 Indexing and access 21
  2.7.1 Numeric indexing 21
  2.7.2 Logical vectors and logical indexing 22
    2.7.2.1 True and False 22
    2.7.2.2 Logical comparisons 22
    2.7.2.3 Logic 22
    2.7.2.4 AND, OR, and NOT 22
    2.7.2.5 Equivalence of TRUE and FALSE to 1 and 0 23
  2.7.3 Two functions that are useul for operating on vectors .. 23
2.8 Assignment 01 24
  2.8.1 The data 24
  2.8.2 Please complete the following 24

2.1 Jupyter Notebooks

All of our lab work will take place in Jupyter Notebooks. Jupyter Notebooks are a tool for organizing textual descriptions of work and computer programs. The goal is to produce one document to communicate a set of scientific ideas and allow another to understand exactly how you arrived at yoru conclusions.

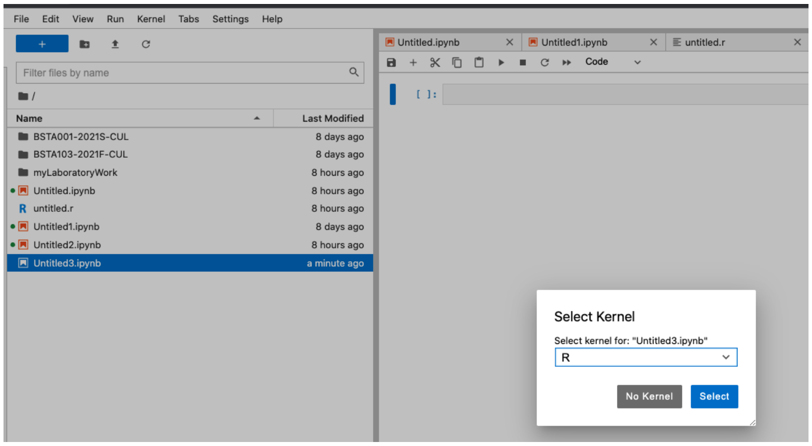
Jupyter has some important buttons.

2.1.1 File

\nobreak

2.1.1.1 A new notebook

Under file->New->Notebook you can create a new notebook. When asked to "Select Kernel" click on the drop down menu and select "R"



Name	Last Modified
BSTA001-2021S-CUL	8 days ago
BSTA103-2021F-CUL	8 days ago
myLaboratoryWork	8 hours ago
Untitled.ipynb	8 hours ago
untitled.r	8 hours ago
Untitled1.ipynb	8 days ago
Untitled2.ipynb	8 hours ago
Untitled3.ipynb	a minute ago

Select Kernel

Select kernel for: "Untitled3.ipynb"

R

2.1.1.2 The notebook

A notebook is a collection of cells. A **cell** is a container that can hold text or computer code. Cells in Jupyter looks like gray rectangles. There are three cell types in Jupyter: (i) Code, (ii) Markdown, and (iii) Raw. The two that we will focus on are Code and Markdown.

The "Code" cell holds computer code that the R kernel (see below about a kernel) can use to compute. We may want to import data, run a statistical analysis, and output results. This is for the "Code" cell.

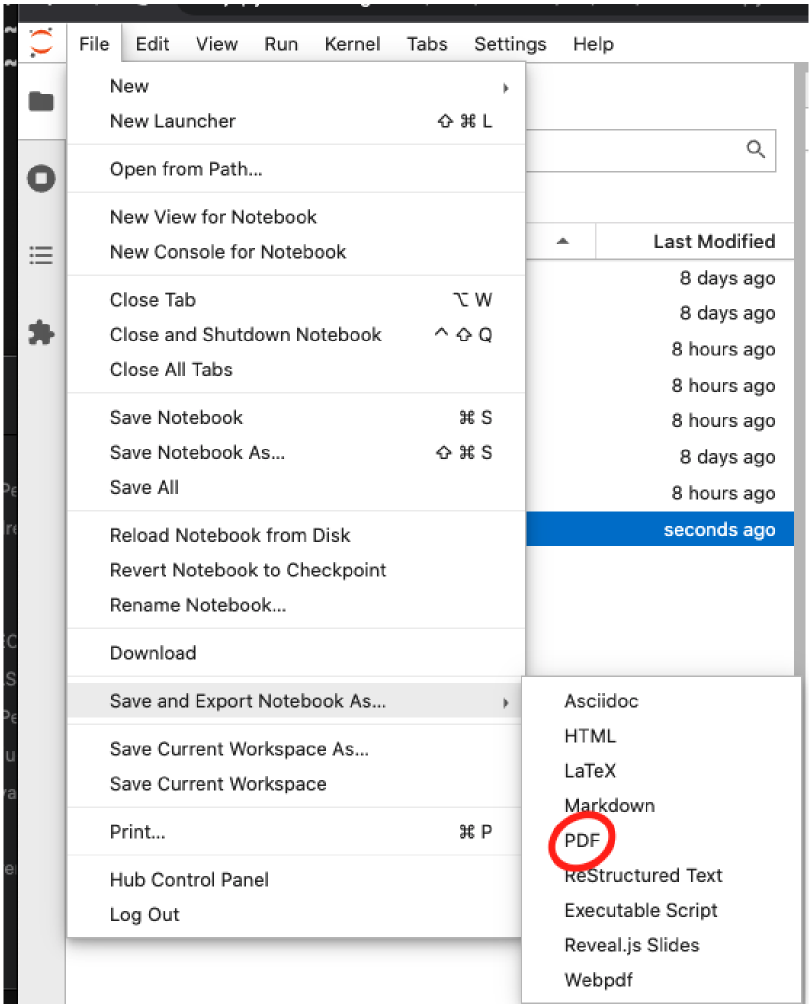
"Markdown" is itself a special language that a Jupyter Notebook interprets as text. The "Markdown" cell is most useful for write ups, descriptions of a Code cell above or below, or scientific conclusions, comments, and thoughts. When you need to write, think Markdown.

2.1.1.3 Save your work

You can always save your work, and should do so often, by clicking File -> Save Notebook.

2.1.1.4 Export for submission

In class, we will ask that you submit your work on Coursesite as a **PDF**. Work in another format will not be accepted. To export your notebook as a PDF, choose File->Save and Export Notebook As->PDF


File
Edit
View
Run
Kernel
Tabs
Settings
Help
New
New Launcher ⇧ ⌘ L
Open from Path...
New View for Notebook
New Console for Notebook
Close Tab ⌥ W
Close and Shutdown Notebook ^ ⇧ Q
Close All Tabs
Save Notebook ⌘ S
Save Notebook As... ⇧ ⌘ S
Save All
Reload Notebook from Disk
Revert Notebook to Checkpoint
Rename Notebook...
Download
Save and Export Notebook As...
Save Current Workspace As...
Save Current Workspace
Print... ⌘ P
Hub Control Panel
Log Out
Asciidoc
HTML
LaTeX
Markdown
PDF
ReStructured Text
Executable Script
Reveal.js Slides
Webpdf
Last Modified
8 days ago
8 days ago
8 hours ago
8 hours ago
8 hours ago
8 days ago
8 hours ago
seconds ago

After you click PDF, a PDF file will be created and saved in a "Downloads" folder on your local machine. Make sure the PDF file contains (1) Your first and surname, the date, and a descriptive title.

2.1.2 Kernel

The kernel is the component that executes code inside your notebook. No kernel, no running code.

Over the course, you may find that your notebook has disconnected or otherwise will no longer execute the code you wrote. Most often, the kernel has stopped. To restart you kernel select Kernel->Restart Kernel.

2.2 Programming and R

The R programming language, while not explictly written for statistics, has a long history as a tool for data analysis, statistics, machine learning, and data science. R supports all of the main paradigms in computing and you will be able to transfer what you learn in R to other programming languages without much difficulty.

Programming is difficult. Like any skill, programming take time to master. Error messages will be commonplace, you will find it difficult to ask the computer to calculate what you want. You will be frustrate and that is ok. Over time you will learn to read the error messages, code will flow more easily. The most important part of programming is daily practice.

When we **execute** code, we ask the computer to translate what we wrote into binary and return a set of results that may or may not be stored in memory. In the Jupyter environment we execute code by pressing "Run" or by using the shortcut "Shift+Enter".

2.3 Arithmetic

R supports all standard artithmetic calculations. Lets "Run" our first computation.

R can interpret addition

[100]:
```
2+2
```

```
4
```

Subtraction

[101]:
```
9-3
```

```
6
```

Division

[102]:
```
3/4
```

```
0.75
```

multiplication

```
[103]: 4*4
```

16

and exponentiation

```
[104]: 3^9
```

19683

As expected, we can compute more difficult arithemetic expressions.

```
[105]: (2^4)+3/2 - 1
```

16.5

2.4 Vectors

The **vector** is the fundemental object in R.

A mathematical vector is an ordered list of numbers. They are denoted by a sequence of numbers surrounded by square brackets.

$$v = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \tag{2.1}$$

Above, the vector **v** is a vector of length 3 and contains, in order, the values 1, 2, and 3.

In R, vectors are goven a name and stored in the computer in one of two ways: (i) using the **c()** operator or (ii) using the assign function.

2.4.1 Assignment

\nobreak

2.4.1.1 c()

We can store a vector named v with the values 1,2,3 in R as follows

```
[106]: v = c(1,2,3)
```

2.4.1.2 assign

We can also use the assign function to store a vector, named q, with the values 3,2,1 as follows

```
[107]: assign("q",c(3,2,1))
```

2.4.1.3 equals

The equals sign **does not** represent two objects are equal to one another. The equals sign in compiuter programming stands for "assign".

When we write `v = c(1,2,3)`, this is understood as "we assign the variable v to the vector (1,2,3). As an example, lets create a vector `(4,5,6)` names `x` and then assign the variable `y` to be the same as `x`

```
[108]: x = c(4,5,6)
       y = x
```

The last line above does not ask whether or not `x` is the same as `y`. Instead, this line assigns the variable `y` to be the same vecor as `x`.

2.5 Print

When we created the vectors **v** and **q** "nothing happened". Though the vector v and q were created and stored in the computer, R does not display these on your screen by default. One way to view any object in R is to print it.

You can print an object, `x`, R by writing `print(x)`

```
[109]: print(v)
```

```
[1] 1 2 3
```

```
[110]: print(q)
```

```
[1] 3 2 1
```

```
[111]: print(x)
       print(y)
```

```
[1] 4 5 6
[1] 4 5 6
```

You do not need to print any object, ever. Printing is not necessary. You should use print to explore whether you programmed something write or to communicate scientific results.

2.6 Combining vectors

We can append one vector to another in R by using the c() operator. Suppose we wish to combine the two vectors

$$x = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} ; z = \begin{bmatrix} -1 \\ 0.2 \\ 90 \end{bmatrix} \tag{2.2}$$

into one vector

$$r = \begin{bmatrix} 1 \\ 2 \\ 3 \\ -1 \\ 0.2 \\ 90 \end{bmatrix} \tag{2.3}$$

Lets first create the vectors `x` and `z`

[112]:

```
x = c(1,2,3)
z = c(-1,0.2,90)
```

Now we can create the vector `r`

[113]:

```
r = c(x,z)
```

If we want to check our work, we can print out `r`.

[114]:

```
print(r)
```

```
[1]  1.0  2.0  3.0 -1.0  0.2 90.0
```

2.7 Indexing and access

Vectors are useful for storing several different numbers. We can access single elements, or several elements inside a vector by (i) naming the vector we want to access, (ii) typing square brackets "[]".

2.7.1 Numeric indexing

If we want to access the 4th element in `r`, we can type

```
[115]: r[4]
```

-1

If we want to access, the 2nd, 4th, and then the first element of `r` we can include in square brackets the vector `c(2,4,1)`

```
[116]: r[c(2,4,1)]
```

1. 2 2. -1 3. 1

We can access the 1st,2nd, and 3rd elements in `r` using the vector `c(1,2,3)`, however a shortcut is to use the **colon** operator. The colon operator takes as input two integers (a,b) separated by a colon (a:b) and expands to the vector `c(a,a+1,a+2,a+3,...,b)`.

Watch

```
[117]: z = 3:5
```

```
[118]: print(z)
```

```
[1] 3 4 5
```

The colon operator is useful for accessing items in a vector

```
[119]: r[2:5]
```

1. 2 2. 3 3. -1 4. 0.2

The above is called **numeric indexing**. Numeric indexing is the access of elements in an object (here a vector) by inputting a single number, or vector of numbers. Indices are always integers. Fractional or decimal numbers cannot be used as indices. Up until now we only used *positive* integers to access elements of a vector.

R also accepts negative integers as indices. Warning: R handles negative indices different than the majority of other progamming languages. A negative index in R standard for **exclude**.

For example, if we want to return all the elements of a vector `q = c(1,4,6,10,0.5)` except for the 2nd element, we can write `q[-2]`

[120]:

```
q = c(1,4,6,10,0.5)
q[-2]
```

1. 1 2. 6 3. 10 4. 0.5

2.7.2 Logical vectors and logical indexing

\nobreak

2.7.2.1 True and False

R, like all programming languages, understands how to operate with binary logic (aside: binary logic is not the only type. If interested, google the tetralemma). True in R is represeted as the word `TRUE` in all capitals. False in R is represented as the word `FALSE` in all capitals. The symbols `TRUE` and `FALSE` are reserved, special symbols in R. You cannot assign a variable to `TRUE` or `FALSE`.

[121]:

```
TRUE
```

TRUE

[122]:

```
FALSE
```

FALSE

2.7.2.2 Logical comparisons

R understand the following logical operators: - > “Greater than” - >= “Greater than or equal to” - < “Less than” - <= “Less than or equal to” - == “Is equal to” - != “Not equal” - `|` “OR” - `&` “AND”

2.7.2.3 Logic

Logic is a method to evaluate statements, sometimes called propositions as either True or False. The above symbols are used to evaluate statements.

When you pose a proposition to R, such as `v > -1` R will evaluate that propositon for each individual element in the vector `v`. Lets create the vector `v = c(-10,10,4)` and ask R to evaluate the proposition `v>-1`.

[123]:
```
v = c(-10,10,4)
v > -1
```

1. FALSE 2. TRUE 3. TRUE

We see that R returns a vector with the same numebr of elements as in v containing the values TRUE or FALSE. A vector that contains values TRUE/FALSE is called a **logical vector**. Like any other vector we can store a logical vector.

[124]:
```
log = v>-1
```

[125]:
```
print(log)
```

```
[1] FALSE  TRUE  TRUE
```

2.7.2.4 AND, OR, and NOT

AND, OR, and NOT are **logical operators**, they allow us to combine one or more propositions. Given two propositions p_1 and p_2, the AND, OR, and NOT operator will evaluate to the following

p_1	p_2	p_1 AND p_2	p_1 OR p_2	$!p_1$
TRUE	TRUE	TRUE	TRUE	FALSE
TRUE	FALSE	FALSE	TRUE	FALSE
FALSE	TRUE	FALSE	TRUE	TRUE
FALSE	FALSE	FALSE	FALSE	TRUE

Logical operators come in handy in **logical indexing**. When we write `r[l]` where `l` is a logical vector, R will return the the values in `r` where `l` is TRUE.

[126]:
```
r = c(-10,0,10,55,0.34,-0.97)
l = c(TRUE,FALSE,TRUE,FALSE,TRUE,TRUE)

r[l]
```

1. -10 2. 10 3. 0.34 4. -0.97

More often we will include the logical statement directly inside the square brackets

[127]:
```
r[ r>0 ]
```

1. 10 2. 55 3. 0.34

2.7.2.5 Equivalence of TRUE and FALSE to 1 and 0

The symbol `TRUE` in R is understood to be the same as the value `1`, and the symbol `FALSE` in R is understood to be the same value as `0`.

[128]:
```
TRUE==1
```

TRUE

[129]:
```
FALSE==0
```

TRUE

[130]:
```
TRUE==0
```

FALSE

[131]:
```
FALSE==1
```

FALSE

2.7.3 Two functions that are useul for operating on vectors

Functions in mathematics take as input a list of objects and return a unique object. The same is true of functions in programming and so in R.

We can create our own functions in R (this will come later), but R also has a large library of built-in functions that are automatically included once you start R. Two very sueful ones are the `sum` function and the `length` function.

The `sum` function takes as input a vector and returns the sum of each element in the function.

[133]:
```
v = c(3,2,1)
sum(v)
```

6

The `length` function takes as input a vector and returns the number of elements in the vector

[134]:
```
length(v)
```

3

2.8 Assignment 01

We are recruited to track the evolution of an infectious agent for a team of public health officials (PHOs). To support future strategic planning the PHOs want to know the impact of intervention X on increases or decreases in the incidence of this infectious agent. The PHO team collected for each county in their state whether the intervention was enacted, and whether the incidence of case counts of this infectious agent increased or decreased 60 days after the intervention was in place data.

Using R, we will assign probabiltiies to the four events in our sample space { (intervention,raise),(intervention, no raise),(no intervention, raise),(no intervention, no raise) }

2.8.1 The data

In the below cell there is a few lines of code pre-programmed. Please runs this cell below.

This cell will create two vectors.

The first vector is called `intervention_rise` and contains one element for each county that has intervention X collected by the PHO team. An element is the value `1` if there was a rise in incidence for the infectious agent and `0` if there was a fall in incidence.

The second vector is called `nointervention_rise` and contains one element for each county that did not have intervention X collected by the PHO team. An element is the value `1` if there was a rise in incidence for the infectious agent and `0` if there was a fall in incidence.

2.8.2 Please complete the following

1. Use the `length` function to count the number of counties where intervention X took place
2. Use the `length` function to count the number of counties where no intervention X took place
3. Use the `sum` function to count the number of counties that observed a rise in incidence.
4. Use the `sum` function and the `not (!)` operator to count the number of counties that observed a fall in incidence.
5. Use the frequentist approach to compute the probability
    - that an intervention would take place in a county
    - that a rise in incidence was observed in a county
    - of a rise in incidence given a county implemented intervention X

  - of a rise in incidence given a county has not implemented intervention X

6. Use the multiplication rule to compute the probability
   - that an intervention and rise is observed (Hint: P(Intervention) * P(Rise | Intervention))
   - that an intervention and fall is observed
   - that no intervention and rise is observed
   - that no intervention and fall is observed
7. Compute the probability of the below events, assuming intervention and rise/fall. are independent
   - Intervention and Rise
   - Intervention and Fall
   - No Intervention and Rise
   - No Intervention and Fall
8. Do you think intervention X is effective at preventing the spread of our infectious agent?

[178]:

```
#RUN THIS CODE. DO NOT WORRY WHAT IT SAYS.
nums = runif(10^3,0,1)

intervention_rise   = c()
nointervention_rise = c()
for (i in nums){
    if (runif(1)> 0.4){
        if ( i>0.800 ){
           risefall = 1
        } else{risefall=0}
        intervention_rise = c(intervention_rise, risefall)
    }
    else {
        if ( i>0.325 ){
           risefall = 1
        } else{risefall=0}
        nointervention_rise = c(nointervention_rise, risefall)
    }
}
```

Part II

Statistics

Bibliography