MET3220C
Computational Meteorology

Lecture 2 – Basic Statistics

Probability
Basic Laws of Probability
Conditional Probability
Independence
Related Concepts in Programming

Events

- We will often want to assess or make use of the probability that an event occurs. In many applications, we will be interested in the probability of a set of events occurring.
  - The single, indivisible, event is called an elementary event.
  - The set of events is called a compound event.
  - Examples?
- We will discuss how the probabilities of elementary events can be combined to determine the probability of a compound event.
- There are simple rules that can be combined to model quite complex situations.
- Computers are very well suited for these calculations.

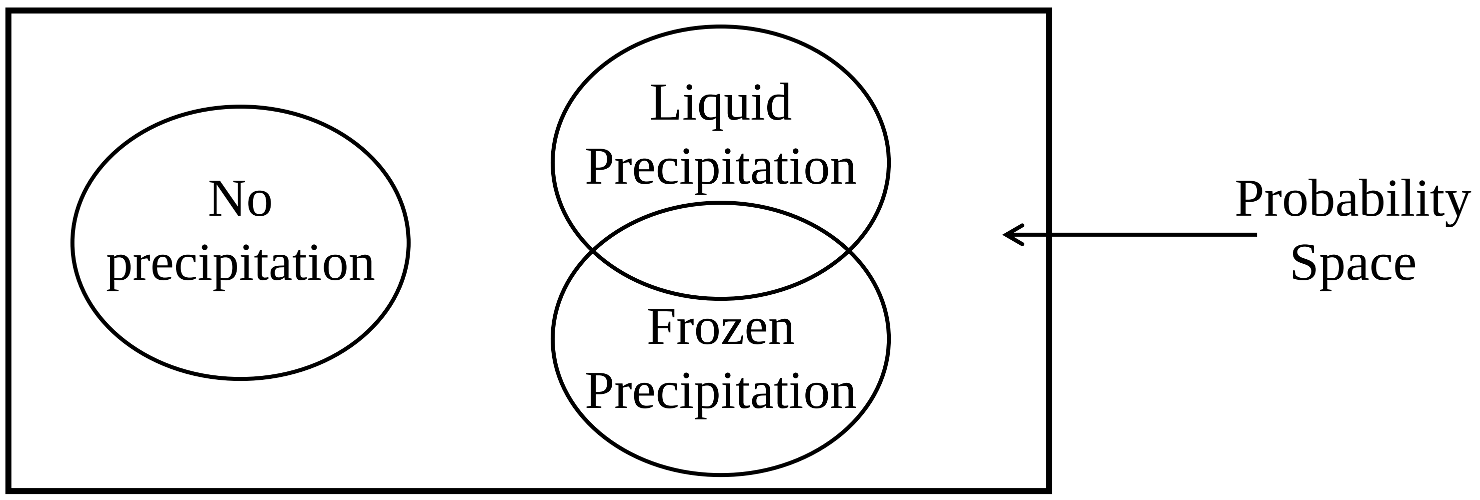
Precipitation Example

- Consider the probability that precipitation occurs (in a day, at a specific location, e.g., my garden).
- Is this an elementary event?
- Is it a compound event made up of chances at several times of day?
  - In reality yes; however, that won't stop us from defining precipitation in terms of daily totals.
- However, meteorologists consider precipitation totals less than 0.01 inches (a 'trace') as not counting as precipitation.
  - Therefore, precipitation means that it liquid or solid water fell, and the daily total was greater than 0.01 inches.
  - A compound event.
- The logic could be further complicated by separately considering different types of precipitation: rain or snow (or hail, or blowing snow, or ….).

Graphical or Spatial Perspective on the Probability of Precipitation

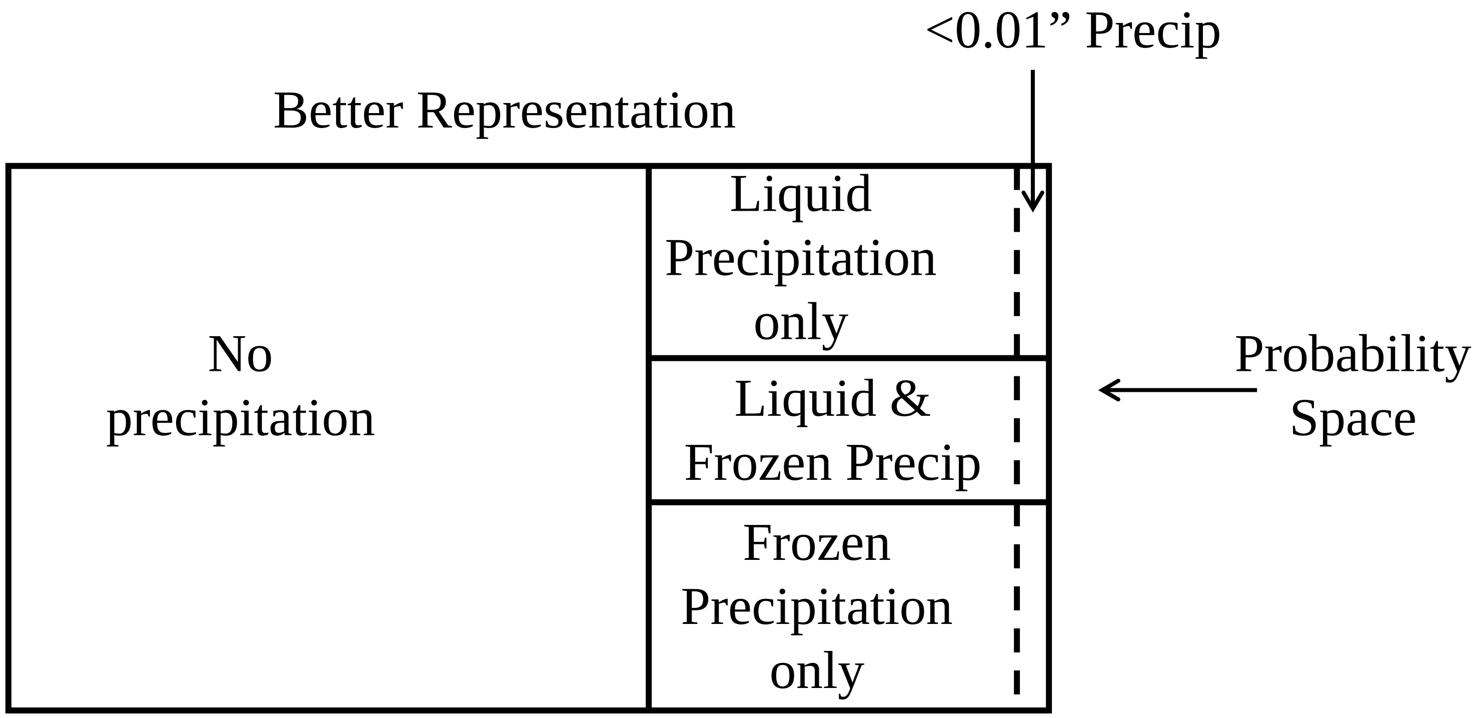
- In (middle school?) math classes the concept is a Venn diagram should have been introduced. A figure is shown drawing the breakdown of outcomes.
- Outcomes that do not overlap in the Venn diagram cannot occur simultaneously.

Traditional Representation



Graphical or Spatial Perspective on the Probability of Precipitation

- If all possibilities are being considered it is most reasonable to have the event span (completely cover) the probability space



- The above diagram could be described as *mutually exclusive and collectively exhaustive* (MECE), made up of 7 elementary events.

Axioms of Probability (The Rules)

- Based on the previous image (or just thinking about the problem), three rules can be used to describe key characteristics of the system and parts of the system.
- (1) The probability of an event is non-negative.
  - In other words, an event cannot occupy negative space in the Venn diagram.
  - Interestingly, a consequence of a negative probability for a physical system would often be that energy could be used to generate more energy.
- (2) The probability of the compound event spanning the system is 1 (100%).
  - Combining rules (1) and (2) indicates the probabilities must be ≤ 1.
- (3) The probability that either of two mutually exclusive events occurs is equal to the sum of the two probabilities of occurrence.
  - Can be extended to apply to any number of such events.

Notation

- Pr{event} means the probability of the event occurring.
    - Example: Pr{rain occurring today) = 0%
    - Example for *Law of Large Numbers*:
        - $\Pr\{|a / n - \Pr\{E\}| < \varepsilon\} \to 0$ as $n \to \infty$,
        - Where *E* is an event, *a* is the number of occurrences, *n* is the number of opportunities, and ε is a arbitrarily small number.
- $\cup$ Union
    - Pr{ event1 $\cup$ event2 } means the probability of either event1 or event2 occurring.
    - Example Pr{ no rain $\cup$ rain } = 1 = 100%
- $\cap$ Intersection
    - Describes the events that belong to both of two categories.
    - Pr{ event1 $\cap$ event2 } is the probability of an event belonging to both categories.
        - If both events are elementary events Pr{ event1 $\cap$ event2 } = 0

Complimentary

- Complimentary is a word that is often used incorrectly by scientists. In regular language, the concept of complimentary is well described by the word *not*.
  - Examples:
    - The compliment of *working* is *not working*.
    - The compliment of *passing* is *failing*.
- The notation for the compliment of an event is event^C.
- If Pr{event} = x, then Pr{event^C} = 1 – x
  - everything in the system space that does not belong to the event

Programming in FORTRAN

- Programming in FORTRAN follows the way people would state the problem rather than the compact statistical notation.
- The intersection of two events would normally be said as both event1 and event2 occurring.
- However, in programming languages, probabilities are either *true* or *false*. This is binary logic.
    - Alternatively, more complicated models can be developed (but are harder to code) with *fuzzy logic*, where fractions can be used
- We would program whether or not an event is part of the intersection as

```
logical :: event1, event2, combo        ! declare the variables
combo = event1 .AND. event2
```

    - To make this practical we would want to have values for event1 and event2 prior to determining the intersection.

Programming in FORTRAN

- The union of two events would normally be said as either event1 or event2 occurring.
- We would program whether or not an event is part of the union as

```
logical :: event1, event2, combo        ! declare the variables
combo = event1 .OR. event2
```

  - To make this practical we would want to have values for event1 and event2 prior to determining the intersection.
- Compliments are very easily programmed

```
logical :: event1, compliment
event1 = .FALSE.      ! Alternatively, we could use .TRUE.
compliment = .NOT. event1
```

De Morgan's Laws

- Augustus De Morgan (a British mathematician), worked with George Boole to modern symbolic logic.
  - Used as the basis for philosophical arguments
  - Used in computers
    - We will skip most of this, and assume that the computer has been correctly programmed
- $(A \cap B)^C = A^C \cup B^C$
- $(A \cup B)^C = A^C \cap B^C$
- Or in FORTRAN
  - .NOT. (A .AND. B) .EQV. .NOT. A .OR. .NOT. B
  - .NOT. (A .OR. B) .EQV. .NOT. A .AND. .NOT. B
  - Why did we use .EQV. rather than = ?

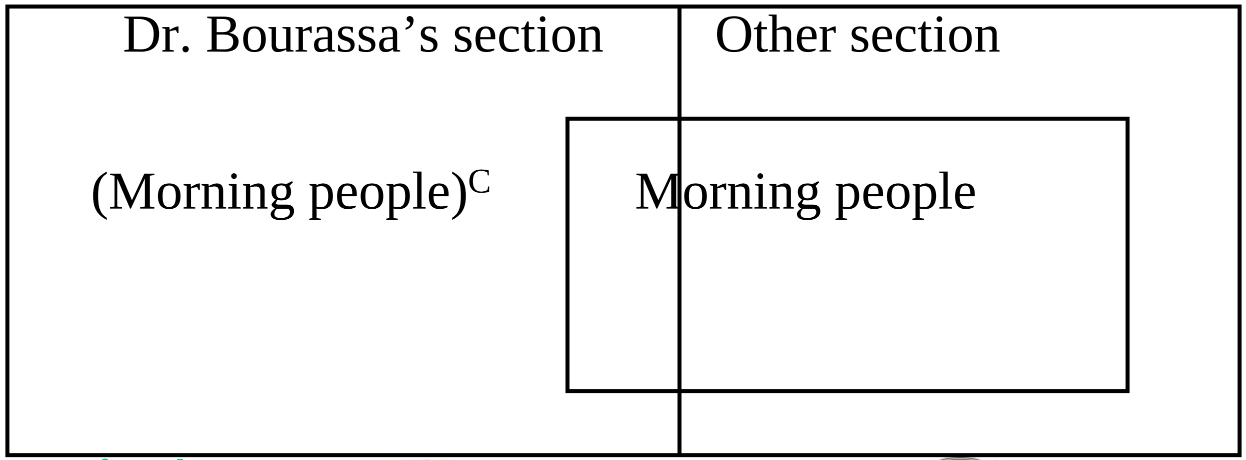
Independence

- Independence is a very very important concept in statistics.
- Events are independent if they are not related
    - What does this mean in terms of the intersection of two independent events?
    - Pr{ event1 $\cap$ event2 } = 0
- What about the probability of the union of the two events?
    - Pr{ event1 $\cup$ event2 } = Pr{ event1 } + Pr{ event2 }

- When we get to statistics involving correlation and uncertainty, the number of independent observations will be a big deal.

Conditional Probability

- The probability of event A, given that event B has occurred, is conditional on the occurrence of event B.
    - Pr{ A | B }
- If event B has occurred, the Pr{ A | B } = Pr{ A }
- Conditional probability can also be thought of in terms of intersection
    - Pr{ A | B } Pr{ B } = Pr{ A ∩ B }
    - Therefore Pr{ A | B } = Pr{ A ∩ B } / Pr{ B }
        - Provided that Pr{ B } ≠ 0



Probability space of MET3220C students

Conditionals in FORTRAN

- When programming, it is often useful to be able to have the code do one thing if an event is true (.TRUE.) and something else if an event is false (.FALSE.).
  - This is rather easy to code with an IF statement.
- Example with words

```
Test1 = .TRUE.  !true if the professor is bald (otherwise false).
IF ( test1 ) THEN
     do not sit in the front row   !avoid being blinded by reflection
ELSE
     sit in front row and give appearance of excitement    !get better grade
ENDIF
```

More Complicated Example

- Often, we have to consider more than one condition.
  - In this case, consider whether or not the professor is bald, and if the professor uses a projector (rather than only writing on the board)
- Example with words

```
test1 = .TRUE.  !true if the professor is bald (otherwise false).
test2 = .TRUE.  !the professor uses a projector at least some of the time
IF ( test2 ) THEN
    IF ( test1 ) THEN
        do not sit in the front row   !avoid being blinded by reflection
    ELSE
        sit in front row and give appearance of excitement    !get better grade
    ENDIF
ELSE
    sit in front row and give appearance of excitement    !get better grade
ENDIF
```

Better Programmed Example

- Often, we have to consider more than one condition.
  - In this case, consider whether or not the professor is bald, and if the professor uses a projector (rather than only writing on the board)
- Example with words

```
test1 = .TRUE.  !true if the professor is bald (otherwise false).
test2 = .TRUE.  !the professor uses a projector at least some of the time
IF ( test2 .AND. test1 ) THEN
    do not sit in the front row   !avoid being blinded by reflection
ELSE
     sit in front row and give appearance of excitement    !get better grade
ENDIF
```

- Could the condition be simplified using De Morgan's rules?

A Word of Advice

- If statements are rather time consuming compared to other commands.
- If the code runs fast anyways, it does not matter.
- However, if the code has many if statements inside DO loops, which are further embedded inside DO loops (and so on), then having too many IF statements can greatly slow a program.