Sequences – have patterns; possible patterns are:
1. Term related to other terms
2. Term described to position in sequence (eg: 1st, 2nd, 3rd)
3. Multiples of 2, 10, etc.

Notation:
\[ a_1 = 1^{st} \text{ term} \]
\[ a_2 = 2^{nd} \text{ term} \]
\[ a_3 = 3^{rd} \text{ term} \]
Recursive formula – term defined in relation to the previous term (write out the above 25 times)

\[ a_5 = a_{5-1} + 2 \] or in notation \[ a_n = a_{n-1} + 2 \]

Closed formula – term defined by relation to its position (write out the above 25 times)

\[ a_n = n \times 2; \quad a_3 = 3 \times 2; \quad a_5 = 5 \times 2 \]

To help discover a sequence, build a table:

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>1</td>
<td>3</td>
<td>7</td>
<td>15</td>
</tr>
</tbody>
</table>

delta = 2 4 8 (differences between numbers in sequence)

Focus on the delta for clues! How is it growing?

Recursive: \[ 2a_{n-1} + 1 \] Closed: \[ 2^n - 1 \]

Some common formulas: \( 2n, 3n, 4n, 2n+1, 3n+1, 4n+1, 2n+2, 3n+2, 4n+2, 2^n+1, 3^n+1, 4^n+1, 2n-1, 3n-1, 4n-1, 2n-2, 3n-2, 4n-2 \)

\( n! \rightarrow n \) factorial - \( 3! = 3 \times 2 \times 1 \)
$0! = 1$

Why?

$1! = 1 \times 0!$

Divide both sides by $1$:

\[
\frac{1!}{1} = \frac{1 \times 0!}{1}
\]

\[
1 = 0!
\]

therefore:

$1 = 0!$
Summation – sum of the terms in a sequence
(write the above 25 times)

\[1+2+3+4+5+6=21\]

Notation:

\[\sum_{i=1}^{n} i\]

Summation symbol tells you that you are to sum terms of sequence

Example:

\[\sum_{i=1}^{5} i+2\]

\[3+4+5+6+7=25\]
Logic

A statement can be either True or False

3 major operations:
- And (Λ): p Λ q
- Or (V): p V q
- Implication (->): p -> q

**AND table**

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>pΛq</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Hint: both must be true in order for statement to be True; all others result in False

**OR Table**

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>pVq</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Hint: ONLY one must be true in order for statement to be True
Implication table

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>p-&gt;q</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

The Rule is: if the phone rings, we MUST pick it up. If the phone rings, and we do NOT pick it up, the whole statement is False. All others are true (because we do not violate the rule). Write out 10 times.

**Connective Not** (essentially reverses the statement to the opposite)

<table>
<thead>
<tr>
<th>p</th>
<th>¬p</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
</tr>
</tbody>
</table>

Tautology – always leads to a true result
Contradiction – always leads to a false result

When faced with a longer logic statement such as:

\( p \rightarrow (p \lor \neg q) \)

break it into parts in a table: \( p, q, \neg q, p \lor \neg q, p \rightarrow (p \lor \neg q) \)

Do parentheses first
To build a 3 element table:
Insert a standard T & F table 2 times:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>

Then add all True for the 1\textsuperscript{st} four lines, then all False for the last 4:

<table>
<thead>
<tr>
<th>p</th>
<th>q</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>T</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
</tbody>
</table>
Sets

A set is a group of objects with no duplicates, separated by commas:
   \{1,2,3,4\}

A set can contain other sets:
   \{1,2,\{cameron, bob\}, 3, 4\}

Sets can be empty:
   \{\}  

Order does not matter:
   \{4,3,6,1,9\}

Can name a set:
   \text{A} = \{1,2,4,16,256\}

Cardinality – refers to the size of the set:
   \text{A} = \{x,y,z\}  
   |\text{A}| = 3 \text{ (cardinality of A is 3)}

\text{E} = \{\{a,b\},\{c,d\}\}  
   |\text{E}| = 2 \text{ (each set is an element; 2 elements)}

Infinite Sets

"N" - natural numbers: \{0,1,2,3...\infty\}
"Z" - integers: \{-2,-1,0,1,2\}
"Q" - rational numbers: numbers that can be expressed as a fraction: \frac{1}{3}, \frac{1}{4}, \frac{1}{5}
"R" - real numbers; all numbers, fractions, decimals
A = \{x/2: x=y*2 \land y={-2,-1,0,1}\}

Read from right to left: put numbers into the y formula, then put results ("x") into x formula.

Cartesian product:

\[ A \times B \]

Take each element in 1\textsuperscript{st} set and combine it with each element in 2\textsuperscript{nd} set, eg: \{1,2\} \times \{4,5\} = \{(1,4),(1,5),(2,4),(2,5)\}
Operations with Sets

Union: $A \cup B$ means to join 2 sets together; no duplicates
$A = \{c,d\}$ $B = \{x,y\}$ $A \cup B = \{c,d,x,y\}$

Intersection: $A \cap B$ – where 2 sets meet or have common values
$A = \{c,d,e,f\}$ $B = \{e,f,g,h\}$
$A \cap B = \{e,f\}$

Notation:

- $\subseteq$: means subset of a larger set
- $\in$: means element of a set

$x : x = 2k & k \in \{1,2,3\} = \{2,4,6\}$ – $x$ is made up of $2k$ and the elements of set $k$ is $\{1,2,3\}$; therefore, $2*1=2$, $2*2=4$, $2*3=6$

Difference between sets:
$A - B$ : one takes the elements of $B$ and removes them from $A$.
$A = \{1,2,3,4\}$ $B = \{2,4\}$ $A - B = \{1,3\}$
Power Sets

Power sets are sets that have taken steroids. 😊
Power sets include all the subsets of the main set, including the empty set:

\[ A = \{1,2,3\} \]
\[ P(A) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\} \]

Another way to indicate a Power set is: \(2^A\)

Cartesian Product
A set that contains all pairs of coordinates:
\[ A = \{1,2\} B = \{y,z\}; A \times B = \{(1,y), (1,z), (2,y), (2,z)\} \]

Set Difference
\[ A-B = \text{takes elements of Set B and removes them from Set A} \]
Functions
In order to be a function, all results of the rule must be in the codomain.

Function Properties:
1. onto: each value in the codomain can be produced by a value in the domain
2. one-to-one: one and only one value in the Domain points to one value in the Codomain

Inverses
The inverse of a function is another function that reverses the process of the original function.

To create an inverse:
- Make the old Codomain the new domain
- Make the old domain the new Codomain
- Swap the f(x) and the x in the formula
- Use algebra to get the f(x) back by itself
Function Composition (ο)
Chains together two functions. 
\[ f \circ g \] which reads “f compose g”.
The result of the second function is plugged into the first function. Another way of writing it is: \( f(g(x)) \); start on the inside and work outwards:

\[ f(g(x)) = f(x) \cdot g(x) \]

Take result of \( g(x) \) and put it into the “x” in \( f(x) \)

---

**Probability**

- **Set of Outcomes**
  This is the set of outcomes that we are interested in occurring.
- **Sample Space**
  This is the set of every possible outcome.

---

- **Set of Outcomes**
  **Sample Space**
<table>
<thead>
<tr>
<th>Duplicates?</th>
<th>Order?</th>
<th>Yes</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
<td>Ordered List ( n^r ) n = size of the list r = # ways to select elements</td>
<td>Unordered List ( \frac{P(n+r-1,r)}{r!} )</td>
</tr>
<tr>
<td>No</td>
<td>Permutation ( P(n,r) = \frac{n!}{(n-r)!} )</td>
<td>Set ( C(n,r) = \frac{P(n,r)}{r!} )</td>
<td></td>
</tr>
</tbody>
</table>
Multiplying Single Dimension Arrays

Licensed under the GNU Free Documentation License (GFDL)
http://www.gnu.org/copyleft/fdl.html

\[
\begin{bmatrix}
10 & 15 & 30
\end{bmatrix}
\begin{bmatrix}
3 \\
4 \\
1
\end{bmatrix}
= 30 + 60 + 30
= 120
\]

\[
\begin{bmatrix}
10 & 3
\end{bmatrix}
\begin{bmatrix}
15 & 4
\end{bmatrix}
\begin{bmatrix}
30 & 1
\end{bmatrix}
= 90 + 30
= 120
\]
Multiplying Single Dimension Matrices

\[
\begin{bmatrix}
2 & 4 & 16 & 32 & 64
\end{bmatrix} \bullet \begin{bmatrix}
64 \\
32 \\
16 \\
4 \\
2
\end{bmatrix} = [2 \times 64] + [4 \times 32] + [16 \times 16] + [32 \times 4] + [64 \times 2] \\
= [128] + [128] + [256] + [128] + [128] \\
= [256] + [256] + [256] \\
= [512] + [256] \\
= [768]
\]
\[
\begin{array}{c}
\begin{array}{cccc}
-2 & -4 & 16 & 32 \\
\downarrow & & \downarrow & \downarrow \\
\text{32} & -16 & -4 & 2 \\
\end{array}
\end{array}
\]

\[
= [-2 \times -32] + [-4 \times -16] + [16 \times -4] + [32 \times 2] \\
= [64] + [64] + [-64] + [64] \\
= [128]
\]

\[
\begin{array}{c}
\begin{array}{ccc}
-3 & -1 & -2 \\
\downarrow & \downarrow & \downarrow \\
2 & 1 & 3 \\
\end{array}
\end{array}
\]

\[
= [-3 \times 2] + [-1 \times 1] + [-2 \times 3] \\
= [-13]
\]

\[
\begin{array}{c}
\begin{array}{cccc}
-1 & 3 & -2 & 5 \\
\downarrow & \downarrow & \downarrow & \downarrow \\
-6 & 9 & -2 & 4 \\
\end{array}
\end{array}
\]

\[
= [-1 \times -6] + [3 \times 9] + [-2 \times -2] + [5 \times 4] \\
= [6] + [27] + [4] + [20] \\
= [37] + [20] \\
= [57]
\]