MATH7501 Practical 3, Semester 1-2021

Topic: Matrices, Sets Counting and Cardinality

Author: Aminath Shausan
Date: 12-03-2021

Pre-Tutorial Activity

- Students must have familiarised themselves with units 1 to 3 contents (Matrices, Sets) of the reading materials for MATH7501

Resources

- https://reference.wolfram.com/language/guide/OperationsOnSets.html

Q 1. Cardinality, Member, Subset and Powerset of a Set

Suppose A = {1, 2, {3}}. Determine if the following statements are true or false.

In[∘]:=
```
Clear[A]
```

In[∘]:=
```
A = {1, 2, {3}}
```

Out[∘]=
```
{1, 2, {3}}
```

(a) $|2^A| = 2^{|A|}$. Use may use **Length[ ]** function to compute cardinality (total number of elements) of a set and **Subsets[ ]** function to compute all subsets of A (which is the powerset)

In[∘]:=
```
Subsets[A]
Length[Subsets[A]]
Length[Subsets[A]] == 2^Length[A]
```

Out[∘]=
```
{{}, {1}, {2}, {{3}}, {1, 2}, {1, {3}}, {2, {3}}, {1, 2, {3}}}
```

Out[∘]=
```
8
```

Out[∘]=
```
True
```

(b) $\{3\} \subset A$. (Use **SubsetQ[ ]** function)

In[∘]:=
```
SubsetQ[A, {{3}}]
SubsetQ[A, {3}]
```

Out[∘]= True

Out[∘]= False

(c) $\{1,2\} \in 2^A$. (Use **MemberQ[ ]** function)

In[∘]:=
```
MemberQ[Subsets[A], {1, 2}]
```

Out[∘]= True

In[∘]:=
```
Clear[A]
```

Q 2. Set Intersection, Union and Complement

Suppose $A = \{k \in \mathbb{N} \mid k \text{ is divisible by } 2\}$ and $B = \{k \in \mathbb{N} \mid k+1 \text{ is divisible by } 2\}$
(a) what is the set $A \cap B$ (ie. the common elements in both A and B)

$A = \{k \in \mathbb{N} \mid k = 2\,n, \text{ for integer } n\}$, $B = \{k \in \mathbb{N} \mid k = 2\,n - 1, \text{ for integer } n\}$. For $n \in \mathbb{N}$,
$A = \{2, 4, 6, 8, 10 \ldots.\}$
$B = \{1, 3, 5, 7, 9 \ldots.\}$
So $A \cap B = \{\,\}$
If considering the following finite sets, you can use **Intersection[ ]** function to compute intersection of A and B

In[∘]:=
```
A = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20};
B = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19};
```

In[∘]:=
```
Intersection[A, B]
```

Out[∘]= { }

(b) what is the set $\mathbb{N} \backslash (A \cup B)$

$A \cup B$ is the set which contains all elements in A and B. So
$A \cup B = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, \ldots\} = \mathbb{N}$. Thus, the complement of $A \cup B$ in $\mathbb{N}$ is
$\mathbb{N} \backslash (A \cup B) = \{\,\}$
Considering the finite sets for A and B you can use the **Union[ ]** and **Complement[ ]** functions to compute union of sets and complement of a set.

In[∘]:=
```
Nat = Table[n, {n, 20}]
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}
```

In[∘]:=
```
AunionB = Union[A, B]
Complement[Nat, AunionB]
```

Out[∘]= {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20}

Out[∘]= { }

(c) If Cr= {k ∈ ℕ| k ≲ r} , determine |A ⋂ C|, for r = 10, 20, using simulation.
The function lenAintC[ ] below computes this. EvenQ[ ] checks if a number is even or not. You can perform the same operation using Divisible[ ] function.

```
ClearAll[r, Cr, A, B]
```

```
lenAintC[r_] := Module[{A, Cr, AinterC},
                                    Cr = Table[k, {k, r}];
                                    A = Select[Cr, EvenQ];
   (*this selects even numbers in Cr*)
                                    (* A = Select[Cr, Divisible[#,2]&];*)

   AinterC = Intersection[A, Cr]; Print[A];
                                    Length[AinterC]
                  ]
```

In[167]:=
```
lenAintC[10]
```

{2, 4, 6, 8, 10}

Out[167]= 5

Q 3. Estimating Least Square Regression Coefficients

Consider matrix A and vector y below.

In[∘]:= $A = \begin{pmatrix} 1 & 3.4 \\ 1 & 3.7 \\ 1 & 5.9 \\ 1 & 6.9 \\ 1 & 9.1 \end{pmatrix}; y = \begin{pmatrix} 2.1 \\ 3.7 \\ 5.8 \\ 8.6 \\ 6.4 \end{pmatrix};$

(a) Generate a million random entries in [0, 5] ×[0,5] for $\beta = [\beta_0, \ \beta_1]^T$, and estimate β which minimises $||A\beta - y||$.
The function lineFit[ ] below performs this simulation.

In[•]:=
```
lineFit[nrun_] := Module[{beta, error, norm, n = nrun, est},
  beta = RandomReal[{0, 5}, {2, n}];
  norm = ConstantArray[0, n];
  For[i = 1, i < n + 1, i++,
   error = A.beta[[All, i]] - y;
   norm [[i]] = Norm[error];
   (*Print[error];*) (* comment*)
  ];
     (*Print[norm];*)
    min = Min[norm]; Print[min];
    indx = Position[norm, min][[1, 1]]; Print[indx];
    (*Print[beta];*)
    beta[[All, indx]]
 ]
```

In[•]:=
```
lineFit[100 000]
```

3.1679

4989

Out[•]= {0.520032, 0.82554}

So the estimates for β is about $(0.520032,0.82554)^{\mathsf{T}}$ using 100,000 runs.

(b) Find the best β using $\beta = (A^{\mathsf{T}} A)^{-1} A^{\mathsf{T}} y$

In[•]:=
```
Inverse[Aᵀ.A].Aᵀ.y // MatrixForm
```

Out[•]//MatrixForm=

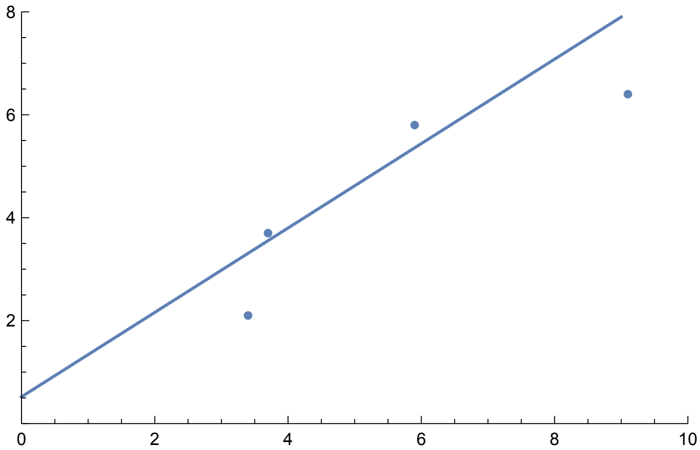
$$\begin{pmatrix} 0.54307 \\ 0.823609 \end{pmatrix}$$

(c) Plot the points and line of best fit

```
xx  = A[[All, 2]];
yy = y[[All, 1]]; (* (xx, yy) are the data points*)
```

```
data = ListPlot[Thread[{xx, yy}], PlotRange → {{0, 10}, {0, 8}}];
(*this plots the data*)
line = Plot[0.52 + 0.82 x, {x, 0, 9}];
(*this plots the line of best fit using the estimates for β*)
Show[data, line]
```

Out[•]=



Q 4. Binomial Coefficient and Non-Integer Solutions

(a) Find all solutions of $x_1 + x_2 + x_3 = 2$

First find the number of solutions using the binomial coefficient

$\binom{n+k-1}{k}$, with $n = 3$,

$k = 2$. You may use **Binomial**[] functionto compute this.

In[•]:=
```
n = 3; k = 2;
```

In[•]:=
```
Binomial[n + k - 1, 2]
```

Out[•]= 6

Now find all solutions.

You may use any of the following two methods to compute all solutions of the above equation

Method 1: Use the **IntegerPertitions[ ]** function to first compute partition of 2 into three parts whose sum is 2, without replacement. The result will be a list of three tuples. Then use the **Permutations[ ]** functions to generate a list of all possible permutations of the elements in the partitioned list.

```
l1 = IntegerPartitions[2, {3}, Range[0, 2]]
l2 = Map[Permutations, l1] (* Map[f,expr] applies
 f to each element on the first level in expe *)
Flatten[l2, 1] (* this flattens the
 nested list l2 at level 1 *)
```

```
Out[•]= {{2, 0, 0}, {1, 1, 0}}

Out[•]= {{{2, 0, 0}, {0, 2, 0}, {0, 0, 2}}, {{1, 1, 0}, {1, 0, 1}, {0, 1, 1}}}

Out[•]= {{2, 0, 0}, {0, 2, 0}, {0, 0, 2}, {1, 1, 0}, {1, 0, 1}, {0, 1, 1}}
```

You can generalise Method 1 for any integers n and k using a Module[ ] function. The function solveLinear[ ] function computes all non-negative integer solutions of the equation $x_1 + x_2 + ... + x_n = k$.

```
In[•]:= Clear[l1, l2]
```

```
In[•]:= solveLinear[nVariable_, sum_] :=
 Module[{n = nVariable, k = sum, l1, l2},
  numberSolns = Binomial[n + k - 1, k];
  Print["Number of Solutions = ", numberSolns];
  l1 = IntegerPartitions[k, {n}, Range[0, k]];
  l2 = Map[Permutations, l1];
  solns = Flatten[l2, 1];
  Print["Solutions =", solns]
 ]
```

```
In[•]:= solveLinear[3, 2]
Number of Solutions = 6
Solutions =
 {{2, 0, 0}, {0, 2, 0}, {0, 0, 2}, {1, 1, 0}, {1, 0, 1}, {0, 1, 1}}
```

Method 2: Use the **Compositions[k,n]** function in the **Combinatorica`** package to find the composition of *k* into *n* parts such that the sum of *n* parts equal *k.* To load the package Combinatorica` type '<< Combinatorica`'. There may be a warning message, ignore it.

```
In[•]:= << Combinatorica`
```

... General: Combinatorica Graph and Permutations functionality has been superseded by preloaded functionality. The package now being loaded may conflict with this. Please see the Compatibility Guide for details.

In[•]:=
```
Compositions[2, 3]
```

Out[•]=
```
{{0, 0, 2}, {0, 1, 1}, {0, 2, 0}, {1, 0, 1}, {1, 1, 0}, {2, 0, 0}}
```