02 Combinatorial Analysis

Pei-Chia Lin

October 24, 2016

```
# Include the necessary packages
library(gtools)
library(partitions)
```

```
# Initialize digits
(digits <- c(0:9))
```

```
##  [1] 0 1 2 3 4 5 6 7 8 9
```

1 List, allow repetition

Permutate three out of 10 digits with repetition

```
# List, allow repetition
# gtools::permutations
digits_lists <- permutations(10, 3, digits, repeats.allowed = TRUE)
head(digits_lists, 50)
```

```
##       [,1] [,2] [,3]
##  [1,]    0    0    0
##  [2,]    0    0    1
##  [3,]    0    0    2
##  [4,]    0    0    3
##  [5,]    0    0    4
##  [6,]    0    0    5
##  [7,]    0    0    6
##  [8,]    0    0    7
##  [9,]    0    0    8
## [10,]    0    0    9
## [11,]    0    1    0
## [12,]    0    1    1
## [13,]    0    1    2
## [14,]    0    1    3
## [15,]    0    1    4
## [16,]    0    1    5
## [17,]    0    1    6
## [18,]    0    1    7
## [19,]    0    1    8
## [20,]    0    1    9
## [21,]    0    2    0
## [22,]    0    2    1
## [23,]    0    2    2
## [24,]    0    2    3
## [25,]    0    2    4
```

```
## [26,]    0    2    5
## [27,]    0    2    6
## [28,]    0    2    7
## [29,]    0    2    8
## [30,]    0    2    9
## [31,]    0    3    0
## [32,]    0    3    1
## [33,]    0    3    2
## [34,]    0    3    3
## [35,]    0    3    4
## [36,]    0    3    5
## [37,]    0    3    6
## [38,]    0    3    7
## [39,]    0    3    8
## [40,]    0    3    9
## [41,]    0    4    0
## [42,]    0    4    1
## [43,]    0    4    2
## [44,]    0    4    3
## [45,]    0    4    4
## [46,]    0    4    5
## [47,]    0    4    6
## [48,]    0    4    7
## [49,]    0    4    8
## [50,]    0    4    9
```

```
# the number of possible outcomes
nrow(digits_lists)
```

```
## [1] 1000
```

```
# the number of possible outcomes = 10*10*10
10^3
```

```
## [1] 1000
```

2 List, allow repetition

The first two elements are selected from the set "letters[1:2]", the middle three elements are selected from the set "digits[1:3]", and the last three elements are selected from the set "letters[24:26]".

```
# List, allow repetition
# base::expand.grid
(sample_lists <- expand.grid(letters[1:2], digits[1:3], letters[24:26]))
```

```
##    Var1 Var2 Var3
## 1     a    0    x
## 2     b    0    x
## 3     a    1    x
## 4     b    1    x
## 5     a    2    x
```

```
## 6     b    2    x
## 7     a    0    y
## 8     b    0    y
## 9     a    1    y
## 10    b    1    y
## 11    a    2    y
## 12    b    2    y
## 13    a    0    z
## 14    b    0    z
## 15    a    1    z
## 16    b    1    z
## 17    a    2    z
## 18    b    2    z
```

```
# the number of possible outcomes
nrow(sample_lists)
```

```
## [1] 18
```

```
# the number of possible outcomes = 2*3*3
2*3*3
```

```
## [1] 18
```

3 Permutation, no repetition, order matters

Permutate three out of 10 digits without repetition

```
# Permutation, no repetition
# gtools::permutations
digits_perm <- permutations(10, 3, digits, repeats.allowed = FALSE)
head(digits_perm, 50)
```

```
##       [,1] [,2] [,3]
##  [1,]    0    1    2
##  [2,]    0    1    3
##  [3,]    0    1    4
##  [4,]    0    1    5
##  [5,]    0    1    6
##  [6,]    0    1    7
##  [7,]    0    1    8
##  [8,]    0    1    9
##  [9,]    0    2    1
## [10,]    0    2    3
## [11,]    0    2    4
## [12,]    0    2    5
## [13,]    0    2    6
## [14,]    0    2    7
## [15,]    0    2    8
## [16,]    0    2    9
## [17,]    0    3    1
```

```
## [18,]    0    3    2
## [19,]    0    3    4
## [20,]    0    3    5
## [21,]    0    3    6
## [22,]    0    3    7
## [23,]    0    3    8
## [24,]    0    3    9
## [25,]    0    4    1
## [26,]    0    4    2
## [27,]    0    4    3
## [28,]    0    4    5
## [29,]    0    4    6
## [30,]    0    4    7
## [31,]    0    4    8
## [32,]    0    4    9
## [33,]    0    5    1
## [34,]    0    5    2
## [35,]    0    5    3
## [36,]    0    5    4
## [37,]    0    5    6
## [38,]    0    5    7
## [39,]    0    5    8
## [40,]    0    5    9
## [41,]    0    6    1
## [42,]    0    6    2
## [43,]    0    6    3
## [44,]    0    6    4
## [45,]    0    6    5
## [46,]    0    6    7
## [47,]    0    6    8
## [48,]    0    6    9
## [49,]    0    7    1
## [50,]    0    7    2
```

```
# the number of possible outcomes
nrow(digits_perm)
```

```
## [1] 720
```

```
# the number of possible outcomes = 10!/(10-3)! = 10*9*8
factorial(10)/factorial(10-3)
```

```
## [1] 720
```

4 Combination, no repetition, order ignored

Select three out of 10 digits without repetition

```
# Combination, no repetition
# gtools::combinations
digits_comb_wo_repeat <- combinations(10, 3, digits, repeats.allowed = FALSE)
head(digits_comb_wo_repeat, 50)
```

```
##       [,1] [,2] [,3]
##  [1,]    0    1    2
##  [2,]    0    1    3
##  [3,]    0    1    4
##  [4,]    0    1    5
##  [5,]    0    1    6
##  [6,]    0    1    7
##  [7,]    0    1    8
##  [8,]    0    1    9
##  [9,]    0    2    3
## [10,]    0    2    4
## [11,]    0    2    5
## [12,]    0    2    6
## [13,]    0    2    7
## [14,]    0    2    8
## [15,]    0    2    9
## [16,]    0    3    4
## [17,]    0    3    5
## [18,]    0    3    6
## [19,]    0    3    7
## [20,]    0    3    8
## [21,]    0    3    9
## [22,]    0    4    5
## [23,]    0    4    6
## [24,]    0    4    7
## [25,]    0    4    8
## [26,]    0    4    9
## [27,]    0    5    6
## [28,]    0    5    7
## [29,]    0    5    8
## [30,]    0    5    9
## [31,]    0    6    7
## [32,]    0    6    8
## [33,]    0    6    9
## [34,]    0    7    8
## [35,]    0    7    9
## [36,]    0    8    9
## [37,]    1    2    3
## [38,]    1    2    4
## [39,]    1    2    5
## [40,]    1    2    6
## [41,]    1    2    7
## [42,]    1    2    8
## [43,]    1    2    9
## [44,]    1    3    4
## [45,]    1    3    5
## [46,]    1    3    6
## [47,]    1    3    7
## [48,]    1    3    8
## [49,]    1    3    9
## [50,]    1    4    5
```

```r
# the number of possible outcomes
nrow(digits_comb_wo_repeat)
```

```
## [1] 120
```

```
# the number of possible outcomes = 10!/(10-3)!3! = (10*9*8)/(3*2*1)
choose(10,3)
```

```
## [1] 120
```

5 Combination, allow repetition

Select three out of 10 digits with repetition

```
# Combinations with repetition
# gtools::combinations
digits_comb_w_repeat <- combinations(10, 3, digits, repeats.allowed = TRUE)
head(digits_comb_w_repeat, 50)
```

```
##       [,1] [,2] [,3]
##  [1,]    0    0    0
##  [2,]    0    0    1
##  [3,]    0    0    2
##  [4,]    0    0    3
##  [5,]    0    0    4
##  [6,]    0    0    5
##  [7,]    0    0    6
##  [8,]    0    0    7
##  [9,]    0    0    8
## [10,]    0    0    9
## [11,]    0    1    1
## [12,]    0    1    2
## [13,]    0    1    3
## [14,]    0    1    4
## [15,]    0    1    5
## [16,]    0    1    6
## [17,]    0    1    7
## [18,]    0    1    8
## [19,]    0    1    9
## [20,]    0    2    2
## [21,]    0    2    3
## [22,]    0    2    4
## [23,]    0    2    5
## [24,]    0    2    6
## [25,]    0    2    7
## [26,]    0    2    8
## [27,]    0    2    9
## [28,]    0    3    3
## [29,]    0    3    4
## [30,]    0    3    5
## [31,]    0    3    6
## [32,]    0    3    7
## [33,]    0    3    8
## [34,]    0    3    9
```

```
## [35,]    0    4    4
## [36,]    0    4    5
## [37,]    0    4    6
## [38,]    0    4    7
## [39,]    0    4    8
## [40,]    0    4    9
## [41,]    0    5    5
## [42,]    0    5    6
## [43,]    0    5    7
## [44,]    0    5    8
## [45,]    0    5    9
## [46,]    0    6    6
## [47,]    0    6    7
## [48,]    0    6    8
## [49,]    0    6    9
## [50,]    0    7    7
```

```
# the number of possible outcomes

# [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
# move from left to right
# two actions: select (S) or shift right (->)
# 000: S S S -> -> -> -> -> -> -> -> ->
# 001: S S -> S -> -> -> -> -> -> -> ->
# ...
# 999: -> -> -> -> -> -> -> -> -> S S S
nrow(digits_comb_w_repeat)
```

```
## [1] 220
```

```
# the number of possible outcomes = (3+9)!/(3!9!)
factorial(3+10-1)/(factorial(3)*factorial(10-1))
```

```
## [1] 220
```

6 Partition

Divide 9 children into 3 teams of 3 each. How many different divisions? (3 teams are unordered)

```
# Partition
# number of group
n_g <- 3
# number of children in each group
n_g1 <- 3
n_g2 <- 3
n_g3 <- 3
# partitions::setparts
(team_unordered_partitions <- setparts(c(n_g1,n_g2,n_g3)))
```

```
##
```

```
##  [1,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [2,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [3,] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 2 2 2 2 2
##  [4,] 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 2 3 3 3 2 2 2 2 2 2 2 2 2 2 1 1 1 1
##  [5,] 3 3 3 3 3 3 2 2 2 2 2 3 3 3 3 3 3 2 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 3
##  [6,] 3 2 2 3 3 2 3 3 3 2 3 2 2 3 3 2 3 3 3 2 3 3 3 2 2 2 3 3 3 2 3 2 2 2
##  [7,] 2 2 3 2 3 3 3 2 3 3 2 3 2 3 2 3 3 3 2 3 3 2 2 3 3 2 3 3 2 3 3 3 3 2
##  [8,] 2 3 2 3 2 3 3 3 2 3 1 1 1 1 1 1 1 1 1 2 3 2 3 2 3 3 2 3 3 2 3 2 3
##  [9,] 1 1 1 1 1 1 1 1 1 1 2 2 3 2 3 3 3 2 3 3 2 2 3 2 3 3 2 3 3 3 2 2 3 3
##
##  [1,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [2,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [3,] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [4,] 1 1 1 1 1 1 3 3 3 3 2 2 2 2 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3
##  [5,] 3 3 2 2 2 2 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2
##  [6,] 3 3 3 3 2 3 3 2 2 2 3 3 2 3 3 3 1 1 1 1 1 1 1 1 1 3 2 2 3 3 2 3 3
##  [7,] 2 2 3 3 3 2 3 3 3 2 3 3 3 2 2 2 3 2 2 3 3 2 3 3 3 2 1 1 1 1 1 1 1 1
##  [8,] 3 2 3 2 3 3 2 3 2 3 3 2 3 3 3 2 2 3 2 3 2 3 3 3 2 3 2 3 2 3 2 3 3 3
##  [9,] 2 3 2 3 3 3 2 2 3 3 2 3 3 3 2 3 2 2 3 2 3 3 3 2 3 3 2 2 3 2 3 3 3 2
##
##  [1,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [2,] 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [3,] 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [4,] 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 1 1 1 1 1 1 1 1 1 1 3 3
##  [5,] 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 1 1
##  [6,] 3 2 3 2 2 3 2 3 3 3 2 3 2 2 3 3 2 3 3 3 2 3 2 2 2 3 3 3 3 2 3 3 2 2
##  [7,] 1 1 2 2 3 2 3 3 3 2 3 3 3 2 2 3 3 2 3 3 3 2 3 3 2 3 2 2 3 3 3 2 3 2
##  [8,] 2 3 2 3 2 3 3 2 3 3 3 2 1 1 1 1 1 1 1 1 1 3 2 3 2 3 2 3 3 2 3 3 3
##  [9,] 3 3 1 1 1 1 1 1 1 1 1 1 2 3 2 2 3 3 3 2 3 3 2 3 3 2 2 3 2 3 3 3 2 3
##
##  [1,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [2,] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [3,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 2 2 2
##  [4,] 3 3 2 2 2 2 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 1 1 1 1 1 1
##  [5,] 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 2
##  [6,] 2 3 3 2 3 3 3 3 1 1 1 1 1 1 1 1 1 1 2 2 3 3 2 3 3 3 2 3 3 2 2 3 2 3
##  [7,] 3 3 3 3 3 2 2 2 2 2 3 3 2 3 3 3 2 3 1 1 1 1 1 1 1 1 1 2 2 3 2 3 3
##  [8,] 2 2 3 3 2 3 3 2 3 2 2 3 3 2 3 3 3 2 3 2 2 3 3 2 3 3 3 2 2 3 2 3 3 3
##  [9,] 3 2 2 3 3 3 2 3 2 3 2 2 3 3 3 2 3 3 2 3 2 2 3 3 3 2 3 3 1 1 1 1 1 1
##
##  [1,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [2,] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [3,] 2 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 2 2 2 2 3 3 3 3
##  [4,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [5,] 3 2 2 2 3 3 3 3 2 3 3 2 2 2 1 1 1 1 1 1 1 1 1 3 3 3 3 2 3 3 2 2 2
##  [6,] 3 3 2 3 2 2 3 2 3 3 3 3 2 3 2 2 2 3 2 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1
##  [7,] 3 2 3 3 3 2 3 3 3 2 2 3 3 2 3 2 3 3 3 2 3 3 2 2 2 2 3 2 3 3 3 3 2 3
##  [8,] 2 3 3 2 1 1 1 1 1 1 1 1 1 1 3 3 2 3 3 3 2 2 3 2 3 2 3 3 3 2 2 3 3 2
##  [9,] 1 1 1 1 2 3 2 3 3 3 2 2 3 3 2 3 3 2 3 3 3 2 2 3 2 3 2 3 3 3 2 2 3 3
##
##  [1,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [2,] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [3,] 3 3 2 2 2 2 3 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2
##  [4,] 1 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 2 2 2 2 3 3 3 3 3
```

```
##  [5,] 3 3 3 3 2 3 3 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [6,] 2 2 3 2 3 3 3 3 2 3 3 2 2 3 2 3 3 3 2 3 2 2 3 2 3 3 3 2 3 3 1 1 1 1
##  [7,] 1 1 1 1 1 1 1 1 1 1 2 2 3 2 3 3 3 2 3 3 3 2 3 3 2 3 3 3 2 2 2 2 3 2
##  [8,] 3 2 3 3 3 2 2 3 3 2 2 3 2 3 3 2 3 3 3 2 1 1 1 1 1 1 1 1 1 3 2 3 3
##  [9,] 2 3 2 3 3 3 2 2 3 3 1 1 1 1 1 1 1 1 1 2 3 2 3 3 3 2 3 3 2 2 3 2 3
##
##  [1,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [2,] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [3,] 2 2 3 3 3 3 3 2 2 2 2 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3
##  [4,] 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 2 2 2 2 3 3 3 3 3 2 2 2 2
##  [5,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 3 3 3 2 3 3 3 2 2 2 3 3 3 2 3 3 3 2
##  [6,] 1 1 1 1 1 1 2 2 3 2 3 3 3 2 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [7,] 3 3 3 2 3 3 1 1 1 1 1 1 1 1 1 1 2 2 3 3 3 2 3 3 2 3 2 3 2 3 3 3 2 3
##  [8,] 2 3 3 3 2 2 3 2 3 3 2 3 3 3 2 2 2 3 2 3 3 3 2 3 3 2 1 1 1 1 1 1 1 1
##  [9,] 3 3 2 3 3 2 2 3 2 3 3 3 2 3 3 2 1 1 1 1 1 1 1 1 1 2 2 3 3 3 2 3 3
##
##  [1,] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##  [2,] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [3,] 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 2
##  [4,] 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 3 3 3 2 2 2 2 3 3 3 3
##  [5,] 2 2 3 3 3 2 3 3 3 2 2 2 3 3 3 2 3 3 3 2 2 2 3 3 3 2 3 3 3 2 2 2 3 3
##  [6,] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 3 2 3 3 3 2 3 3 2 2 3 2 3 3 3 2 3 3 2 2 3
##  [7,] 3 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
##  [8,] 1 1 2 3 2 3 3 3 2 3 3 2 2 2 3 3 3 2 3 3 2 3 1 1 1 1 1 1 1 1 1 1 1 1
##  [9,] 2 3 2 2 3 3 3 2 3 3 2 3 1 1 1 1 1 1 1 1 1 2 2 3 3 3 2 3 3 2 3 1 1
##
##  [1,] 1 1 1 1 1 1 1 1
##  [2,] 2 2 2 2 2 2 2 2
##  [3,] 2 2 2 3 3 3 3 3
##  [4,] 3 3 2 2 2 2 3 3
##  [5,] 3 2 3 3 3 2 2 2
##  [6,] 2 3 3 3 2 3 3 2
##  [7,] 3 3 3 2 3 3 2 3
##  [8,] 1 1 1 1 1 1 1 1
##  [9,] 1 1 1 1 1 1 1 1
```

```
# the number of possible outcomes
ncol(as.matrix(team_unordered_partitions))
```

```
## [1] 280
```

```
# the number of possible outcomes = (9!/(3!3!3!))/3!
(factorial(n_g1+n_g2+n_g3)/(factorial(n_g1)*factorial(n_g2)*factorial(n_g3)))/factorial(n_g)
```

```
## [1] 280
```