R Notebook

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(combinat)
```

```
##
## Attaching package: 'combinat'
```

```
## The following object is masked from 'package:utils':
##
##     combn
```

Designing Process Layouts

```r
sideX <- 3
sideY <- 2
layoutX <- 3
layoutY <- 2
cells <- (layoutX * layoutY)
dt <- data.frame(
  as.matrix(Matrix::forceSymmetric(matrix(c(0, 30, 0, 34, 50, 25,
                                            NA, 0, 0, 55, 10, 10,
                                            NA, NA, 0, 0, 15, 5,
                                            NA, NA, NA, 0, 0, 0,
                                            NA, NA, NA, NA, 0, 30,
                                            NA, NA, NA, NA, NA, 0),
                                          cells, cells), uplo = "L")))

colnames(dt) <- LETTERS[1:cells]

# perms <- lapply(permn(LETTERS[1:cells]), function(x) {
#   matrix(x, layoutY, layoutX)
#   })
let <- c(1:cells)
names(let) <- LETTERS[1:cells]

perms <- list(matrix(c("F", "B", "E", "C", "A", "D"), 2, 3),
              matrix(c("C", "B", "D", "E", "A", "F"), 2, 3))

pathsLet <- t(combn(names(let), 2))
pathsN <- t(combn(1:cells, 2))

totals <- unlist(lapply(perms, function(x) {
  sum(apply(pathsLet, 1, function(y) {
    start <- which(x == y[1], arr.ind = TRUE)
    end <- which(x == y[2], arr.ind = TRUE)
    ((abs(end[,"row"] - start[,"row"]) * (sideY / layoutY)) +
      (abs(end[,"col"] - start[,"col"]) * (sideX / layoutX))) *
      dt[let[y[1]], y[2]]
  }))
}))

min(totals)
```

```
## [1] 419
```

```r
perms[which(totals == min(totals))]
```

```
## [[1]]
##      [,1] [,2] [,3]
## [1,] "F"  "E"  "A"
## [2,] "B"  "C"  "D"
```