Permutation

Gina Reynolds and Matt Winters

This file is used in replicating findings for "Foreign Aid Funnel? Placebo-Based Assessment of Aid Flows to Non-Permanent U.N. Security Council Members" by Evangeline Reynolds (University of Illinois) and Matthew Winters (University of Illinois).

```
library(knitr)
library(stringr)
```

Join UNSC Tenures with Region Data

```
load("03_UNSC - SCTenuresData.RData")

SCTenures=TenuresData
names(SCTenures)
```

```
## [1] "CountryName"           "TwoYearTenure"         "SCYearOne"
## [4] "SCYearTwo"             "StandardName"          "YearAfterSCMembership"
```

```
dim(SCTenures)
```

```
## [1] 315   6
```

```
FirstYear=min(SCTenures$SCYearOne)
LastYear=max(SCTenures$SCYearOne)
NumTenures=dim(SCTenures)[1]
```

```
FrequenciesTimesOnSC=table(table(SCTenures$StandardName))
names(FrequenciesTimesOnSC)=paste("Times on SC =", names(FrequenciesTimesOnSC))
```

We observe non-permanent Security Council tenures with the first year being 1946 to 2014, (the second year of the most recent tenure is 2015). There are a total of 315 country tenures.

```
OneYearTenures=SCTenures[is.na(SCTenures$SCYearTwo),]
NumOneYearTenures=dim(OneYearTenures)[1]
OneYearTenures=OneYearTenures[order(OneYearTenures$SCYearOne),c("StandardName", "SCYearOne")]
colnames(OneYearTenures)=c("Country", "SC One Year Tenure")
rownames(OneYearTenures)=1:dim(OneYearTenures)[1]
```

Normally countries are on the Security Council for two years at a time, however there are 15 cases where the country is on the UNSC for just one year. In the permutations, we do not treat these tenures differently. These one-year tenures exist in the permutations at the historic moment where they existed, and any country that is eligible for assignment in a given year may be assigned to the one-year tenures.

```
# For the time span 1946-1965
NumberOfSeatYears1946_1965=length(1946:1965) * 6
NumberOfSeatYears1966_2014=length(1966:2014) * 10
+ 5 * length(2015)
```

```
## [1] 5
```

```
dim(SCTenures)[1] * 2 - sum(is.na(SCTenures$SCYearOne) + is.na(SCTenures$SCYearTwo))
```

```
## [1] 615
```

For the period 1946 to 1965, there are six non-permanent seats per year for a total of 120 seat-years. For the period 1966 to 2014, there are ten non permanent seats per year for a total of 490 seat-years.

```
load("03_UNRegions - DATAAllRegionsLong.RData")
ls()
```

```
##  [1] "AcceptableDraw"             "AcceptAssignment"
##  [3] "AcceptDraw"                 "Characters"
##  [5] "Country"                    "COUNTRY"
##  [7] "CountryCouncilYears"        "CountryNamesColumns"
##  [9] "CountryRegionsTenure"       "CountryYearData"
## [11] "CountyNamer"                "CurrentYear"
## [13] "DATA"                       "DATAAllRegionsLong"
## [15] "doc.type.out"               "EachCountry"
## [17] "EarlyCountryYear"           "EarlyCountryYearRegions"
## [19] "EarlyDATA"                  "EarlyRegionsData"
## [21] "EarlyUniqueRegions"         "EndDate"
## [23] "FAILURE"                    "FavoriteCountryName"
## [25] "FirstYear"                  "FrequenciesTimesOnSC"
## [27] "i"                          "j"
## [29] "LastYear"                   "LateRegionsData"
## [31] "League"                     "long"
## [33] "LongSCPermutations"         "MinAcceptableDifference"
## [35] "ModernCountryYear"          "ModernCountryYearRegions"
## [37] "ModernDATA"                 "ModernRegions"
## [39] "MyCounter"                  "NumberOfSeatYears1946_1965"
## [41] "NumberOfSeatYears1966_2014" "NumCountries"
## [43] "NumCountryRegions"          "NumDrawsBfrAccptbl"
## [45] "NumIntervals"               "NumOneYearTenures"
## [47] "NumPermutation"             "NUMPROBLEMS"
## [49] "NumTenures"                 "NUMTENURES"
## [51] "NumYears"                   "OneCase"
## [53] "OneYearTenures"             "OriginalNumCountries"
## [55] "p"                          "PermColumns"
## [57] "Permutation"                "PermutationsYr1"
## [59] "PermuteNumCountries"        "Problem"
## [61] "PROBLEMS"                   "q"
## [63] "REGION"                     "RegionsData"
## [65] "SCTenures"                  "StandardName"
## [67] "StartDate"                  "startNumeric"
## [69] "Subset"                     "SUBSET"
```

```
## [71] "SUCCESSES"                  "Temp"
## [73] "TenuresData"                "text"
## [75] "text0"                      "text1"
## [77] "text2"                      "text3"
## [79] "text4"                      "text5"
## [81] "TheDraw"                    "time0"
## [83] "time1"                      "TimesOnSC"
## [85] "TryAgainFreshStart"         "UniqueCountries"
## [87] "UniqueLateRegions"          "UniqueRegions"
## [89] "UNMember"                   "UNMemberData"
## [91] "UNMemLong"                  "Year"
## [93] "Year1"                      "Year2"
## [95] "Year3"                      "YearDraw"
## [97] "YearsInUN"
```

```
RegionsData=DATAAllRegionsLong
names(RegionsData)
```

```
## [1] "StandardName" "Year"         "Region1"      "Region2"
```

```
head(RegionsData)
```

```
##   StandardName Year             Region1 Region2
## 1    Argentina 1946 Latin American Seat    <NA>
## 2    Argentina 1947 Latin American Seat    <NA>
## 3    Argentina 1948 Latin American Seat    <NA>
## 4    Argentina 1949 Latin American Seat    <NA>
## 5    Argentina 1950 Latin American Seat    <NA>
## 6    Argentina 1951 Latin American Seat    <NA>
```

```
tail(RegionsData)
```

```
##       StandardName Year       Region1 Region2
## 10785     Zimbabwe 2010 African Group    <NA>
## 10786     Zimbabwe 2011 African Group    <NA>
## 10787     Zimbabwe 2012 African Group    <NA>
## 10788     Zimbabwe 2013 African Group    <NA>
## 10789     Zimbabwe 2014 African Group    <NA>
## 10790     Zimbabwe 2015 African Group    <NA>
```

```
UniqueRegions=unique(RegionsData$Region1)
```

There are ten regions from which countries are elected to the Security Council. They are different for the periods 1946-1965 and 1966-2014.

```
EarlyRegionsData=RegionsData[RegionsData$Year<1966,]

EarlyUniqueRegions=unique(str_replace(as.character(EarlyRegionsData$Region1)," Seat",""))
```

```
LateRegionsData=RegionsData[RegionsData$Year==1966,]
UniqueLateRegions=unique(c(as.character(LateRegionsData$Region1)))
```

```
head(RegionsData)
```

```
##   StandardName Year             Region1 Region2
## 1    Argentina 1946 Latin American Seat    <NA>
## 2    Argentina 1947 Latin American Seat    <NA>
## 3    Argentina 1948 Latin American Seat    <NA>
## 4    Argentina 1949 Latin American Seat    <NA>
## 5    Argentina 1950 Latin American Seat    <NA>
## 6    Argentina 1951 Latin American Seat    <NA>
```

```
dim(SCTenures)
```

```
## [1] 315   6
```

```
SCTenures=merge(SCTenures, RegionsData, by.x=c("StandardName", "SCYearOne"),
                by.y=c("StandardName", "Year"), all.x=TRUE)
head(SCTenures)
```

```
##   StandardName SCYearOne CountryName TwoYearTenure SCYearTwo
## 1      Algeria      1968     Algeria     1968-1969      1969
## 2      Algeria      1988     Algeria     1988-1989      1989
## 3      Algeria      2004     Algeria     2004-2005      2005
## 4       Angola      2003      Angola     2003-2004      2004
## 5    Argentina      1948   Argentina     1948-1949      1949
## 6    Argentina      1959   Argentina     1959-1960      1960
##   YearAfterSCMembership                             Region1 Region2
## 1                  1970 African Group - League Arab States    <NA>
## 2                  1990 African Group - League Arab States    <NA>
## 3                  2006 African Group - League Arab States    <NA>
## 4                  2005                      African Group    <NA>
## 5                  1950                 Latin American Seat    <NA>
## 6                  1961                 Latin American Seat    <NA>
```

```
tail(SCTenures)
```

```
##     StandardName SCYearOne CountryName TwoYearTenure SCYearTwo
## 310   Yugoslavia      1988  Yugoslavia     1988-1989      1989
## 311       Zambia      1969      Zambia     1969-1970      1970
## 312       Zambia      1979      Zambia     1979-1980      1980
## 313       Zambia      1987      Zambia     1987-1988      1988
## 314     Zimbabwe      1983    Zimbabwe     1983-1984      1984
## 315     Zimbabwe      1991    Zimbabwe     1991-1992      1992
##     YearAfterSCMembership                Region1 Region2
## 310                  1990 Eastern European Group    <NA>
## 311                  1971          African Group    <NA>
## 312                  1981          African Group    <NA>
## 313                  1989          African Group    <NA>
## 314                  1985          African Group    <NA>
## 315                  1993          African Group    <NA>
```

```r
dim(SCTenures)
```

```
## [1] 315   8
```

```r
SCTenures[is.na(SCTenures$Region1),]
```

```
## [1] StandardName          SCYearOne             CountryName
## [4] TwoYearTenure         SCYearTwo             YearAfterSCMembership
## [7] Region1               Region2
## <0 rows> (or 0-length row.names)
```

```r
SCTenures=SCTenures[order(SCTenures$Region1),]

head(SCTenures)
```

```
##     StandardName SCYearOne CountryName TwoYearTenure SCYearTwo
## 14     Australia      1946   Australia     1946-1947      1947
## 15     Australia      1956   Australia     1956-1957      1957
## 56        Canada      1948      Canada     1948-1949      1949
## 57        Canada      1958      Canada     1958-1959      1959
## 114        Ghana      1962       Ghana     1962-1963      1963
## 128        India      1950       India     1950-1951      1951
##     YearAfterSCMembership           Region1 Region2
## 14                   1948 Commonwealth Seat    <NA>
## 15                   1958 Commonwealth Seat    <NA>
## 56                   1950 Commonwealth Seat    <NA>
## 57                   1960 Commonwealth Seat    <NA>
## 114                  1964 Commonwealth Seat    <NA>
## 128                  1952 Commonwealth Seat    <NA>
```

```r
table(SCTenures$Region, useNA="ifany")
```

```
## < table of extent 0 >
```

Turkey

Turkey is a special case because it was a part of two regions in the period 1946-1965. It serves twice on the UNSC as part of the Eastern European and Asia Region, but once, starting in 1951, on the UNSC as a Middle Eastern representative.

```r
SCTenures[SCTenures$StandardName=="Turkey", ]
```

```
##     StandardName SCYearOne CountryName TwoYearTenure SCYearTwo
## 286       Turkey      1951      Turkey     1951-1952      1952
## 287       Turkey      1954      Turkey     1954-1955      1955
## 288       Turkey      1961      Turkey          1961        NA
## 289       Turkey      2009      Turkey     2009-2010      2010
##     YearAfterSCMembership                                Region1
## 286                  1953          Eastern European and Asia Seat
```

```
## 287                  1956           Eastern European and Asia Seat
## 288                  1962           Eastern European and Asia Seat
## 289                  2011 Western European and Others Group (WEOG)
##                 Region2
## 286 Middle Eastern Seat
## 287 Middle Eastern Seat
## 288 Middle Eastern Seat
## 289                <NA>
```

```
dim(SCTenures)
```

```
## [1] 315   8
```

```
SCTenures$Region=SCTenures$Region1
SCTenures$Region[SCTenures$StandardName=="Turkey"&
                    SCTenures$SCYearOne==1951]="Middle Eastern Seat"

SCTenures[SCTenures$StandardName=="Turkey", ]
```

```
##     StandardName SCYearOne CountryName TwoYearTenure SCYearTwo
## 286       Turkey      1951      Turkey     1951-1952      1952
## 287       Turkey      1954      Turkey     1954-1955      1955
## 288       Turkey      1961      Turkey          1961        NA
## 289       Turkey      2009      Turkey     2009-2010      2010
##     YearAfterSCMembership                                 Region1
## 286                  1953           Eastern European and Asia Seat
## 287                  1956           Eastern European and Asia Seat
## 288                  1962           Eastern European and Asia Seat
## 289                  2011 Western European and Others Group (WEOG)
##                 Region2                                    Region
## 286 Middle Eastern Seat                      Middle Eastern Seat
## 287 Middle Eastern Seat           Eastern European and Asia Seat
## 288 Middle Eastern Seat           Eastern European and Asia Seat
## 289                <NA> Western European and Others Group (WEOG)
```

Preliminaries for the Permutations

The permutations are based on the idea of assigning the most constrained countries first.

We calculate a variable for the number of times we observe a country on the UNSC - (number of tenures) - by country and by region (which changes at 1966). This variable is the starting point for keeping track of reassignment. As reassignments are made, we subtract off tenures for individual countries, the remaining tenures, still unassigned will enter into the probability of being assigned or not.\

```
SCTenures$Observation=1
TimesOnSC=with(SCTenures, aggregate(Observation, by=list(StandardName),  FUN=sum))
names(TimesOnSC)=c("StandardName","NumTenures")
head(TimesOnSC)
```

```
##   StandardName NumTenures
## 1      Algeria          3
```

```
## 2       Angola          1
## 3    Argentina          9
## 4    Australia          5
## 5      Austria          3
## 6   Azerbaijan          1
```

```
SCTenures=merge(SCTenures, TimesOnSC, by="StandardName")
dim(SCTenures)
```

```
## [1] 315  11
```

The "Long UNmember Data" allows us to understand the temporal range of possibilities. We will not allow a country's Security Council tenure reassignment to be to a year in which the country is not yet a U.N. member.

```
load("03_UNMemLong.RData")
UNMember=UNMemLong
```

We aggregate on both region and country to get the "RegionCountryNumTenures". We'll sort on this. The greater the number of tenures in a region, the more problematic a country might be in terms of reassignment. We don't want such countries to be crowded out of allowed reassignments - i.e. have other reassignments force them into implausible timings. \

```
options(width=75)
CountryRegionsTenure=with(SCTenures, aggregate(StandardName,
                                by=list(StandardName, Region),
                                FUN="length"))
head(CountryRegionsTenure)
```

```
##       Group.1           Group.2 x
## 1   Australia Commonwealth Seat 2
## 2      Canada Commonwealth Seat 2
## 3       Ghana Commonwealth Seat 1
## 4       India Commonwealth Seat 1
## 5 Ivory Coast Commonwealth Seat 1
## 6 New Zealand Commonwealth Seat 1
```

```
names(CountryRegionsTenure)=c("StandardName", "Region", "RegionCountryNumTenures")
head(CountryRegionsTenure)
```

```
##   StandardName            Region RegionCountryNumTenures
## 1    Australia Commonwealth Seat                       2
## 2       Canada Commonwealth Seat                       2
## 3        Ghana Commonwealth Seat                       1
## 4        India Commonwealth Seat                       1
## 5  Ivory Coast Commonwealth Seat                       1
## 6  New Zealand Commonwealth Seat                       1
```

The number of years on the U.N. also contributes to the level of constraint for a reassignment, this is because the fewer years a country is on the U.N., the fewer years there are available for reassigned.

```
options(width=100)
YearsInUN=aggregate(UNMember$UNMember,
                    by=list(UNMember$StandardName),
                    FUN=sum)
head(YearsInUN)
```

```
##               Group.1  x
## 1         Afghanistan 70
## 2             Albania 61
## 3             Algeria 54
## 4             Andorra 23
## 5              Angola 40
## 6 Antigua and Barbuda 35
```

```
names(YearsInUN)=c("StandardName", "YearsUNMember")
CountryRegionsTenure=merge(CountryRegionsTenure, YearsInUN, by="StandardName")

head(CountryRegionsTenure)
```

```
##   StandardName                                      Region RegionCountryNumTenures YearsUNMember
## 1      Algeria          African Group - League Arab States                       3            54
## 2       Angola                                African Group                       1            40
## 3    Argentina Latin American and Caribbean Group (GRULAC)                       7            71
## 4    Argentina                        Latin American Seat                       2            71
## 5    Australia                          Commonwealth Seat                       2            71
## 6    Australia    Western European and Others Group (WEOG)                       3            71
```

```
#Make Dummy Variable for Early Region
CountryRegionsTenure$EarlyRegion=0
CountryRegionsTenure$EarlyRegion[grep("Seat", CountryRegionsTenure$Region)]=1
CountryRegionsTenure$EarlyRegion
```

```
##   [1] 0 0 0 1 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1
##  [48] 1 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 1 0 0 1 0 1 0 1 0 0 1 1 0 0 0 0 1 1 0 0 0 0 0 1 0 0
##  [95] 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
## [142] 0 1 0 0 0 0 0 1 1 0 1 0 0 1 0 1 1 1 0 0 0 0 1 0 0 0
```

```
CountryRegionsTenure[CountryRegionsTenure$StandardName=="United Arab Republic",]
```

```
##             StandardName              Region RegionCountryNumTenures YearsUNMember EarlyRegion
## 157 United Arab Republic Middle Eastern Seat                       1             4           1
```

```
#To avoid many permutation failures, we order the data.
CountryRegionsTenure=
  CountryRegionsTenure[order(-(CountryRegionsTenure$StandardName=="United Arab Republic"),
                             CountryRegionsTenure$Region,
                             CountryRegionsTenure$YearsUNMember/
                               CountryRegionsTenure$RegionCountryNumTenures),
                       ]

head(CountryRegionsTenure)
```

```
##              StandardName              Region RegionCountryNumTenures YearsUNMember EarlyRegion
## 157 United Arab Republic Middle Eastern Seat                        1             4           1
## 5               Australia   Commonwealth Seat                        2            71           1
## 26                 Canada   Commonwealth Seat                        2            71           1
## 76            Ivory Coast   Commonwealth Seat                        1            56           1
## 55                  Ghana   Commonwealth Seat                        1            59           1
## 138             Sri Lanka   Commonwealth Seat                        1            61           1
```

The Permutations

```
#Long$HighlyConstrained=
table(CountryRegionsTenure$Region)
```

```
##
##                            Commonwealth Seat              Eastern European and Asia Seat
##                                            8                                           10
##                          Latin American Seat                          Middle Eastern Seat
##                                           12                                           10
##                         Western European Seat                           Asia-Pacific Group
##                                            8                                           12
##                        Eastern European Group          African Group - League Arab States
##                                           16                                            9
##    Western European and Others Group (WEOG)                                African Group
##                                           20                                           33
## Latin American and Caribbean Group (GRULAC)     Asia-Pacific Group - League Arab States
##                                           19                                           10
```

```
head(CountryRegionsTenure)
```

```
##              StandardName              Region RegionCountryNumTenures YearsUNMember EarlyRegion
## 157 United Arab Republic Middle Eastern Seat                        1             4           1
## 5               Australia   Commonwealth Seat                        2            71           1
## 26                 Canada   Commonwealth Seat                        2            71           1
## 76            Ivory Coast   Commonwealth Seat                        1            56           1
## 55                  Ghana   Commonwealth Seat                        1            59           1
## 138             Sri Lanka   Commonwealth Seat                        1            61           1
```

```
#Council
NumCountryRegions=dim(CountryRegionsTenure)[1]
NumCountryRegions
```

```
## [1] 167
```

```
time0 <- Sys.time()

# To make this exactly replicable, we set a starting point for the randomization
set.seed(1234)
NumPermutation #=10# the number of permutation may be fed in from control file or you can set it here.
```

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

```
# We are going to attempt making permutation until we have as many successes (SUCCESSES)
# as our desired number (NumPermutation)
# We start off with zero successes, but this is a counter that is added to as we create more
# and more successful permutations
SUCCESSES=0
while (SUCCESSES<NumPermutation){

  #We create an ID variable, this
  SCTenures$ID=1:dim(SCTenures)[1]
  SCTenures$Unfilled=1
  Permutation=rep(NA,dim(SCTenures)[1])

  NumDrawsBfrAccptbl=rep(FALSE,NumCountryRegions)
  AcceptableDraw=rep(FALSE,NumCountryRegions)
  MyCounter=rep(0,NumCountryRegions)
  PROBLEMS=rep(NA,NumCountryRegions)
  COUNTRY=rep(NA,NumCountryRegions)
  NUMTENURES=rep(NA,NumCountryRegions)
  FAILURE=rep(NA,NumCountryRegions)
  REGION=rep(NA,NumCountryRegions)

  # Now we go through each region (REGION), and permute the SC assignment within them.
  try(
  for (i in 1:NumCountryRegions){

      # For each country region, we identify the Country, Region,
      # and Number of Tenures associated with the country
      COUNTRY[i]=as.character(CountryRegionsTenure$StandardName[i])
      #if(i==1){print(COUNTRY[i])}
      REGION[i]=as.character(CountryRegionsTenure$Region[i])
      #if(i==1){print(REGION[i])}
      NUMTENURES[i]=CountryRegionsTenure$RegionCountryNumTenures[i]
      #if(i==1){print(NUMTENURES[i])}

      # We subset to the region of interest, where we like to reassign the timing of the
      # UNSC memberships.
      # Note that we are subseting from the SCTenures, the Dataset which contains an
      # observation for each UNSC tenure.
      SUBSET=SCTenures[SCTenures$Region==REGION[i],]
      #if(i==1){print(dim(SUBSET))}
      SUBSET$SCYearOne
      #if(i==1){print(head(SUBSET$ID))}

      SUBSET$Unfilled

      # We merge on the UN membership status for the country of interest that we've
      # subsetted to.  If the country of interest is not in the UN in a given year
      # we will not let it be assigned to be on the UNSC in that year.
      SUBSET$CountryOfInterest=COUNTRY[i]
      table(SUBSET$CountryOfInterest)
      SUBSET=merge(SUBSET, UNMember, by.x=c("CountryOfInterest", "SCYearOne"), by.y=c("StandardName", "
      names(SUBSET)
```

```
    SUBSET[1:3,]


     SUBSET$UNMember

     NUMTENURES[i]


# Now we assign the year of the tenure.  There are several things to consider.
# Any one country can only fill one seat on the Security Coucil at any one time.
# The randomization should not all reassignment of two or more tenures of one
# country to overlap temporaly in the reassignment.
# Also, we never see consecutive tenures.  There is a minimum two-year break
# before non-permanent members can return to the Council again.

          # Recall that Failure is set to NA; FAILURE=rep(NA,NumCountryRegions)
          # We start off with the presumption that there won't be a failure
          # This will be overwritten if after 10,000 attempts at a good reassignment,
          # there is no acceptable result
          FAILURE[i]=FALSE
          MyCounter=0
          AcceptDraw="Not Acceptable"
          TryAgainFreshStart=FALSE

          while(AcceptDraw=="Not Acceptable"& TryAgainFreshStart==FALSE){
            # Step 1 Start Counting attempts to get an acceptable reassignment
            MyCounter=MyCounter+1



              TheDraw=sample(SUBSET$ID, # From the IDs that correspond to the region
                             NUMTENURES[i], #, draw the number of tenures the
                                      # country in question has been on the
                                      # council-from this region.
                             replace=FALSE, # don't allow the same slot to be drawn more than once
                             prob=SUBSET$Unfilled* # If the routine has already filled
                                                   # up one of the allowable
                                                   # years with another country,
                                                   # then we don't allow it to assign the
                                                   # country to that year (ID)
                               SUBSET$UNMember # Also, if the country in question
                                               # is not a member of the UN in the id year,
                                               # don't allow that assignment
                             )
            # Sometimes all these conditions will not be met given previous assignments,
            # and a sample cannot be taken
            # Then R produces an error "Error in sample.int(length(x), size, replace, prob) :
            # incorrect number of probabilities"- this will break the for-loop.
            # Since we specify try, R begins the premutation creation process again

            # Which are the years associated with the Draw?  TheDraw is a vector of ID numbers
              YearDraw=SCTenures$SCYearOne[TheDraw]
            # Order the years
```

```
      YearDraw=sort(YearDraw)

    # We don't want our permutations to have tenures that are back to back.
    # We define an acceptable Difference in Start Years for SC tenures
    MinAcceptableDifference=3

    # Then we determine if this assumed interval between tenures is met.
    if(length(YearDraw)==1){  # For the case of just one tenure, there is never a problem
        AcceptDraw="Acceptable" # So we overwrite AcceptDraw with "Acceptable"
        }else{ # but for more than one tenure year we have to worry about this spacing.
            NumIntervals=length(YearDraw)-1 # The number of intervals to evaluate is
                                            # the number of start years minus one.
            Problem=rep(NA,NumIntervals)   # We create an empty vector which we will
                                          #populate with information about whether
                                          # there is a problem  with the interval or not.
            for (q in 1:NumIntervals){
              Problem[q]=(YearDraw[q+1]-YearDraw[q])<  # is the difference between start
                          MinAcceptableDifference    #years less than what is allowed
                                # If so Problem is populated with "TRUE"
            }

            NUMPROBLEMS=sum(Problem)  # How many interval problems are observed?

            if(NUMPROBLEMS==0){ # If there are zero problems

              AcceptDraw="Acceptable" # then, the Draw is determined to be "Acceptable"
                                      # This will stop the While loop, which will try again
                                      # with country[i] and region[i]
            } else {
              AcceptDraw="Not Acceptable" # Otherwise it is "Not Acceptable"
                                          # The while loop keeps going - trying to descover
                                          # an acceptable reassignment for country[i]
                                          # in region[i]
            }

          }

    if(MyCounter>1000){
      TryAgainFreshStart=TRUE # We allow the routine to make 10,000 attempts
                              # at assigning an acceptable configuration for country[i]
                              # within its region
                              # If it just doesn't happen,
                              # then we start the permutation routine over from scratch
                              # TryAgainFreshStart=TRUE
                    # Recall that Failure is set to NA; FAILURE=rep(NA,NumCountryRegions)
    FAILURE[i]=TRUE}  # We just keep track of failures observed for a given country-region
  }

#This is for curiousity's sake.
AcceptableDraw[i]=AcceptDraw # For each country[i] and its region,
                             # is there an acceptable draw?
NumDrawsBfrAccptbl[i]=MyCounter-1
```

```
        # Permutation was set to NAs, but as we make successful draws, then we reassign these NAs
        # with the country[i] for the IDs drawn.
        # This assigns the country a new set of years to be on SC, still within its region.
        Permutation[TheDraw]=as.character(COUNTRY[i])

        # This vector keeps track of which replacements have been made already,
        # Recall starting out, SCTenures$Unfilled=1
        # But slowly, these slots get filled, so we overwrite those positions with zero
        # We'll set the probability of drawing an id,
        # associated with a region-year that has been filled, to zero
        SCTenures$Unfilled[TheDraw]=0

}       # End of for loop through each country-region,
        # From here we go back and go again through additional country-regions.

)       # End of try - Ensures that error (in the sampling stage - when allowable slots are blocked)
        # Will not end our attempts to make acceptable permutations

# Recall that Failure is set to NA; FAILURE=rep(NA,NumCountryRegions)
# Now we overwrite FAILURE due to the incomplete assignment in the while loop,
# or because of an error:
# The NA is overwriten with TRUE, there was a failure
    FAILURE[is.na(FAILURE)]=TRUE


if(!any(FAILURE==TRUE)){  #If the run wasn't a failure so far, check one more reason
            # It is possible that countries are still too close to each other
            # Even though we tried to handle this with care, there could still be
            # unforseen overlap when the regions change in 1965 to 1966;
            # also, Turkey is a difficult case, as it is in two regions

            UniqueCountries=unique(Permutation)
            NumCountries=length(UniqueCountries)
            AcceptAssignment=rep(NA,NumCountries)
            for (p in 1:NumCountries){
            Temp=data.frame(SCTenures,Permutation)

                Subset=Temp[Temp$Permutation==UniqueCountries[p],]

                YearDraw=Subset$SCYearOne
                YearDraw=sort(YearDraw)
                #print(YearDraw)

                            # Then we determine if this assumed interval between tenures is met.
            if(length(YearDraw)==1){  # For the case of just one tenure, there is never a problem
                AcceptAssignment[p]="Yep" # So we overwrite AcceptAssignment with "Yep"
                }else{ # but from more than one tenure year, we have to worry about this spacing.
                    NumIntervals=length(YearDraw)-1 # The number of intervals to evaluate is
                                                    # the number of start years minus one.
                    Problem=rep(NA,NumIntervals)   # We create an empty vector which we will
                                                   # populate with information about whether
                                                   # there is a problem with the interval or not.
                    for (q in 1:NumIntervals){
```

```
                        Problem[q]=(YearDraw[q+1]-YearDraw[q])<  # is the difference between start
                                    MinAcceptableDifference    # years less than what is allowed
                                          # If so Problem is populated with "TRUE"
                      }

                      NUMPROBLEMS=sum(Problem)  #How many interval problems are observed?

                      if(NUMPROBLEMS==0){ #If there are zero problems

                        AcceptAssignment[p]="Yep" #then, the Draw is determined to be "Yep"
                                          #This will stop the While loop, which will try again
                                          # with country[i] and region[i]
                      } else {
                        AcceptAssignment[p]="Nope" # Otherwise it is "Nope"
                                                # The while loop keeps going - trying to discover
                                                # an acceptable reassignment for country[i] in
                                                # region[i]
                      }

                    }

             } # ends for loop through p countries - not region specific

   if(any(AcceptAssignment=="Nope")){FAILURE[1:length(FAILURE)]=TRUE}

  } # ends if no failure conditional




 if(any(FAILURE==TRUE)){# print(
      "Permutation Failure"
      #)
      }else{
      ##This is a counter for how many successful permutation have been made.
      ##Recall it starts at zero
      SUCCESSES=SUCCESSES+1

      #We make the permutation vector a column of data associated with the SCTenures
      SCTenures$Permutation=Permutation

      #Rename the variable SCTenures$Permutation to index it:
      #eg: SCTenures$Permutation1, SCTenures$Permutation2 etc
      names(SCTenures)[dim(SCTenures)[2]]=paste("Permutation",SUCCESSES,sep="")

      }

}
```

```
time1 <- Sys.time()
time1-time0
```

```
## Time difference of 9.312621 hours
```

```
#j
options(width=100)

data.frame(AcceptableDraw, NumDrawsBfrAccptbl, COUNTRY, NUMTENURES, REGION)



head(table(Permutation, useNA="ifany"))
head(table(SCTenures$StandardName, useNA="ifany"))

head(SCTenures)

PermColumns=grep("Permutation",names(SCTenures))
PermColumns
sum(is.na(SCTenures[,PermColumns]))

SCTenures=SCTenures[order(SCTenures$Region,SCTenures$SCYearOne),  ]

#write.csv(SCTenures, "05_Wide_Permutations.csv")
#save(SCTenures, file="05_Wide_Permutations.RData")
```

```
names(SCTenures)
PermColumns=grep("Permutation",names(SCTenures))
head(PermColumns)
CountryNamesColumns=c(which(names(SCTenures)=="StandardName"),PermColumns)
head(CountryNamesColumns)
Year1=data.frame(Year=as.numeric(SCTenures$SCYearOne),
                 WhichTenureYear=1,
                 SCTenures[,CountryNamesColumns])
Year2=data.frame(Year=as.numeric(SCTenures$SCYearTwo),
                 WhichTenureYear=2, SCTenures[,CountryNamesColumns] )
Year3=data.frame(Year=as.numeric(SCTenures$YearAfterSCMembership),
                 WhichTenureYear=3, SCTenures[,CountryNamesColumns])

#recall that for the year "3" there are cases where this is really the year after

LongSCPermutations=rbind(Year1, Year2, Year3)
names(LongSCPermutations)

#head(LongSCPermutations[,1:5])

dim(LongSCPermutations)
#Sometimes there is not a second year
# and get rid of permanent members
LongSCPermutations=LongSCPermutations[!is.na(LongSCPermutations$Year), ]
dim(LongSCPermutations)

dim(SCTenures)
dim(LongSCPermutations)
table(LongSCPermutations$WhichTenureYear)
```

```
LongSCPermutations=LongSCPermutations[order(LongSCPermutations$StandardName, LongSCPermutations$Year),]
str(LongSCPermutations)

#Drop 2016
LongSCPermutations=LongSCPermutations[LongSCPermutations$Year!=2016,]
min(LongSCPermutations$Year);max(LongSCPermutations$Year)

names(LongSCPermutations)[names(LongSCPermutations)=="StandardName"]="RealizedAssignment"

save(LongSCPermutations, file="05_LongSCPermutations.RData")
write.csv(LongSCPermutations, "05_LongSCPermutations.csv", row.names=FALSE)
```

Check that Countries Appear on the UNSC with the Same Frequency in the Permutations as in Reality

```
PermutationsYr1=LongSCPermutations[LongSCPermutations$WhichTenureYear==1,]

OriginalNumCountries=table(PermutationsYr1$RealizedAssignment)
PermuteNumCountries=table(PermutationsYr1$RealizedAssignment)

identical(OriginalNumCountries, PermuteNumCountries)
```

```
## [1] TRUE
```

```
for (i in sample(1:NumPermutation, 10, replace=FALSE)){

print(identical(OriginalNumCountries, table(PermutationsYr1[, c(paste0("Permutation", i))])))
  }
```

```
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
## [1] TRUE
```