**Systematic Sampling HW Assistance**

The following process explains the implementation of systematic sampling in R. For this example, estimating Grade is of interest. It is know that DaysSkipped is thought to impact Grade; thus, systematic sampling will be used to obtain a Grade across the entire spectrum of the DaysSkipped variable.

* Step 1: Read-in original data
* Step 2: Sort the data.frame by a variable (DaysSkipped here) that is thought to impact the response variable (Grade here).

|  |  |
| --- | --- |
| Original Data  Data <- read.csv(file.choose()) | Data Sorted by DaysSkipped  library(dplyr)  Data2 <- arrange(Data,DaysSkipped) |

* Step 3: After the data has been sorted, create a RowIndex variable.

|  |  |
| --- | --- |
|  | Adding a unique row index to the data.frame  Data2 <- mutate(Data2,RowIndex = seq(1,37,1)) |

The total number of rows in this data.frame is 37. Suppose the desired sample size is 5.

37 / 5 = 7 r 2

The fact that the math does not come out evenly complicates things a bit. For example, if there were 35 total rows with a desired sample size of 5, then there would be 7 rows in each group. However, for this example, there will be 7 people in most groups, but the last group will have a total of 9 (7 + 2 extra).

|  |  |
| --- | --- |
| Desired sample size = 5  Total Number of Rows = 37  37 / 5 = 7 r 2 | Groups 1 – 4 will each have 7 people  and Group 5 will have (7 + 2 ) people |

* Step 4: Use modular arithmetic to create a repeated pattern for each of the 5 groups.

|  |  |
| --- | --- |
|  | Adding an index that is repeated for each group.  Data3 <- mutate(Data2,ModIndex = RowIndex %% 7) |

* Step 5: Obtain a random integer to identify the row will be used from each group.

|  |  |
| --- | --- |
|  | Adding an index that is repeated for each group.  > sample(0:6,size=1) |

The fact that the last group has 9 (instead of 7) people may result in too many people being selected for the sample. The implication of this is that you may end up with 1 additional observations in your sample – which may or may not be a big deal.

|  |  |
| --- | --- |
| *Works as intended*  Sample Size = 5  > sample(0:6,size=1)  [1] 5 | *Does not work as intended*  Sample Size = 6 (not 5 as intended)  > sample(0:6,size=1)  [1] 2 |

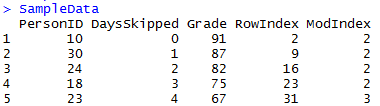
The following procedure can be used to obtain the desired sample size (exactly).

* + Divide the sampling into two pieces: 1) Part 1 - randomly selected rows using the systematic sampling process. The randomly selected rows will be taken from all group except the last group. 2) For the last group, obtain a simple random sample of size 1 from the rows in the last group.

|  |  |  |
| --- | --- | --- |
| PART 1 | Systematic Sampling for Groups 1 – 4 | Part1 <- Data3   %>% filter(RowIndex <= 4\*7)   %>% filter(ModIndex == 2) |
| PART 2 | Simple random sampling for Group 5 | > sample(29:37,size=1)  [1] 31  Part2 <- filter(Data3,RowIndex == 31) |

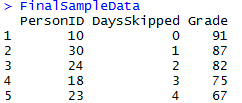
Finally, concatenate the two parts into single data.frame, say SampleData.

> SampleData <- bind\_rows(Part1,Part2)



The following use of the select() function can be used to keep only the columns of interest.

FinalSampleData <- select(SampleData,PersonID,DaysSkipped,Grade)



Finally, compute the statistic of interest.

> summarize(FinalSampleData, "Estimate" = mean(Grade))

Estimate

1 80.4

The following function puts all these pieces together and can be used to accomplish systematic sampling for the data used in this example.

SystematicSample <- function(){

getmodindex <- sample(0:6,size=1)

Part1 <- Data3 %>% filter(RowIndex <= 4\*7) %>% filter(ModIndex == getmodindex)

getlastrowindex <- sample(29:37,size=1)

Part2 <- filter(Data3,RowIndex == getlastrowindex)

SampleData <- bind\_rows(Part1,Part2)

FinalSampleData <- select(SampleData,PersonID,DaysSkipped,Grade)

output <- summarize(FinalSampleData, "Estimate" = mean(Grade))

return(output)

}

Using this function to obtain an estimate for average grade.

> SystematicSample()

Estimate

1 78