Generate Random Observation Times

Samuel P Callisto

July 19, 2018

Function to generate times between 10:30 and 12:30

```
generateObsTime <- function(){
  ## multiplicative factor
  i <- 1
  ## single draw from uniform distribution [0,1]
  r <- runif(1)

  ## iterate i by 1 for each value of 0.2
  while(i <= 5 & r > 0.2*i){
    i <- i + 1
  }

  ## multiply i by half-hour to get random obs time between 10:30 & 12:30
  amObs <- 10 + 0.5 * i

  ## print r & i to screen to check output
  print(paste0("r=", round(r,2), "   i=", i))

  ## return the observation time
  return(amObs)
}
```

Generate random observation times for 10 patients

```
## set random seed for reproducibility
set.seed(3980287)

## number of patients to generate times for
nPatients <- 10

## create container for for-loop output
obsTimes <- vector("double", nPatients)

## generateObsTime for nPatients and save to output container element i
for(i in 1:nPatients){
  obsTimes[[i]] <- generateObsTime()
}
```

```
## [1] "r=0.23   i=2"
## [1] "r=0.44   i=3"
## [1] "r=0.88   i=5"
## [1] "r=0.81   i=5"
## [1] "r=0.58   i=3"
## [1] "r=0.91   i=5"
```

```
## [1] "r=0.35   i=2"
## [1] "r=0.58   i=3"
## [1] "r=0.48   i=3"
## [1] "r=0.04   i=1"
```

```
## display output
obsTimes
```

```
##  [1] 11.0 11.5 12.5 12.5 11.5 12.5 11.0 11.5 11.5 10.5
```

generate a function that you can scale to multiple patients by passing nPatients as an argument

```
generateNObsTimes <- function(nPatients, seed=3980287){
  set.seed(seed)

  ## create container for for-loop output
  obsTimes <- vector("double", nPatients)

  ## generateObsTime for nPatients and save to output container element i
  for(i in 1:nPatients){
    obsTimes[[i]] <- generateObsTime()
  }

  return(obsTimes)
}

generateNObsTimes(10)
```

```
## [1] "r=0.23   i=2"
## [1] "r=0.44   i=3"
## [1] "r=0.88   i=5"
## [1] "r=0.81   i=5"
## [1] "r=0.58   i=3"
## [1] "r=0.91   i=5"
## [1] "r=0.35   i=2"
## [1] "r=0.58   i=3"
## [1] "r=0.48   i=3"
## [1] "r=0.04   i=1"
```

```
##  [1] 11.0 11.5 12.5 12.5 11.5 12.5 11.0 11.5 11.5 10.5
```