**Inputs**

//Vector that describes the month.Coding for the vector is as follows:

Vector should have a day for every day of the month + the last day of previous month + 1st day of next month.

1: Let full work days

2: Days leading into a weekend

3: Weekend days

4: Final day of a weekend.

Let entries from this vector be denoted as VM\_i

V\_month = []

Length(V\_month) = number of days in the month + 2

Monthly\_total = Monthly\_total(V\_month[])

//This is a vector that denotes all personnel availability to be an input to the watchbill generation algorithm. Coding for the vector is as follows:

0: available for duty, no restrictions

1: First day of leave or travel day heading to a TDY trip

2: On leave or TDY

3: Final day of leave or travel day from a TDY trip

4: Day before special liberty (Likely travel)

5: Special liberty. Unable to stand duty but no credit for watchbill purposes.

6: Day after special liberty (night watch only)

7: Local event. Cannot stand watch but does not impact surrounding days

Let entries from this vector be denotes at VPA\_i where i is the day of the month

V\_PersonnelAvailability,v = []

//This is a matrix that has all watchstanders availability vectors in it. Let the rows denote the personnel names and the columns denote the days of the month.

Let entries of this matrix be denotes MA\_i,j where i is the person and j is the day of the month

M\_availability,m = [[]]

**Constants**

// Weighting for cost of 1 hour of duty. Intended to equate the “cost” of a weekend duty day to a work week duty day. Weekend starts Friday at noon and ends sunday at midnight.

WH (Working hour) = 1

WODH (Weekday off duty hour) = 2

WEH (Weekend hour) = 3

//How to account for the :) from missing a day in the office because of night shift

Missed work day benefit = 0

//Night watch is a pain in the ass. Adjustment?

Night watch factor = 0

//Defines the work day

Work = 8 (0800-1600)

//Defines the D shift

Day = (0800-2000) (8 WH + 4 ODH)

//Defines the N shift

Night = 2000-0800 (12 ODH)

**Generic Functions**

//Calculates total amount of shift points for each day of the month

Input = V\_month vector

Output = Monthly\_total []

1: Full work days

2: Days leading into a weekend

3: Weekend days

4: Final day of the weekend.

*Function* Monthly\_total (V\_month[]) ={

Monthly\_total= []

For i in V\_month{

If V\_month[i] ==1:

Monthly\_total[i] = 16\*WODH + 8\*WH

Else if V\_month[i] ==2:

Monthly\_total[i] = 8\*WODH + 8\*WH + 8\*WEH

Else if V\_month[i] == 3 or 4:

Monthly\_total[i] = 24\*WEH

Else:

Print (invalid input given in input vector)

Return Null

}

Return Monthly\_total

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Calculates total amount of shift points for the month:

*Function* Monthly\_totalscore (V\_month[]) ={

Score = Sum(Monthly\_total(V\_month[]))

Return Score

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Calculates total availability score for each person.

Input = V\_PersonnelAvailability,v vector, V\_monthly

Output = Vector that describes when person can stand watch. Most useful as sum to determine expected watch

0: available for duty, no restrictions

1: First day of leave or travel day heading to a TDY trip

2: On leave or TDY

3: Final day of leave or travel day from a TDY trip

4: Day before special liberty (Likely travel)

5: Special liberty. Unable to stand duty but no credit for watchbill purposes.

6: Day after special liberty (night watch only)

7: Local event. Cannot stand watch but does not impact surrounding days

*Function* Personnel\_Availability\_Vector (V\_PersonnelAvailability[], Monthly\_total[]) = {

OutputVector = []

For i in V\_PersonnelAvailability{

If V\_PersonnelAvailability[i] == 0,4,5 or 6:

OutputVector[i] = Monthly\_total[i]

Else if V\_PersonnelAvailability[i] == 1, 2, 3, or 7:

OutputVector[i] = 0

Else:

Print (invalid input given in input vector)

Return Null

}

Return OutputVector

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Creates a vector that shows what watches the person is actually available for

Input =V\_PersonnelAvailability

Output = V\_WatchbillAvailability

Input code:

0: available for duty, no restrictions

1: First day of leave or travel day heading to a TDY trip

2: On leave or TDY

3: Final day of leave or travel day from a TDY trip

4: Day before special liberty (Likely travel)

5: Special liberty. Unable to stand duty but no credit for watchbill purposes.

6: Day after special liberty (night watch only)

7: Local event. Cannot stand watch but does not impact surrounding days

Output code:

0: Not available for any watch

1: Available for Day watch only

2: Available for Night watch only

3: Available for either watch

*Function* WatchbillAvailability(V\_PersonnelAvailability[]) {

V\_WatchbillAvailability = []

For i in V\_PersonnelAvailability {

// Currently out of area or on leave

If V\_PersonnelAvailability[i] == 1, 2, 3, 4, 5, or 7

V\_PersonnelAvailability[i] =0

// Coming off of work travel/leave or going into work travel/leave: no watch

Else if V\_PersonnelAvailability[i-1] == 3 or V\_PersonnelAvailability[i+1] == 1

V\_PersonnelAvailability[i] =0

// Travel for liberty day before, only night watch this day

Else If V\_PersonnelAvailability[i-1] == 6 and V\_PersonnelAvailability[i] == 0

V\_PersonnelAvailability[i] =2

// Travel for liberty day likely next day, only day watch this day

Else If V\_PersonnelAvailability[i] == 0 and V\_PersonnelAvailability[i+1] == 4

V\_PersonnelAvailability[i] =1

Else if V\_PersonnelAvailability[i-1] == 0 && V\_PersonnelAvailability[i] == 0 && V\_PersonnelAvailability[i+1] == 0

V\_PersonnelAvailability[i] =3

Return V\_PersonnelAvailability

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Creates a vector that has expected number of watch points for each qualified watchstander

Input: V\_month[], M\_Availability[[]]

Output: Expected watch per person []

*Function* ExpectedWatchVector(M\_Availability[[]], Monthly\_total[]) {

MonthlyTotalWatch = MonthlyTotalScore(Monthly\_total[])

WatchstandAvailVector = []

CommandAvailability = 0

//Determine each person's availability. Output is a vector of total watch points the commandcan stand this month. Each row is a person and magnitude is availability.

For i in num.rows(M\_Availability[[]])

OutputVector = Personnel\_Availability\_Vector (M\_Availability[i,[]], Monthly\_total[])

WatchstandAvailVector[i] = Sum(OutputVector[])

CommandAvailability = sum(WatchstandAvailVector[])

//Multiply WatchstandAvailVector by the MonthlyTotalWatch and divide by Command Availability. This will give an estimated “weighted number” of watch each person should stand.

WatchstandAvailVector[] = (WatchstandAvailVector[]/CommandAvailability) \* MonthlyTotalWatch

Return WatchstandAvailVector

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Turns the M\_Availability[[]] into a usable matrix that says whether each person can stand watch. Each row will be a watchstander and each column is a day of the month with the corresponding output code.

Output code:

0: Not available for any watch

1: Available for Day watch only

2: Available for Night watch only

3: Available for either watch

*Function* WatchstanderAvailability(M\_Availability[[]]){

WatchstanderAvailability = [[]]

For i in numrows (M\_Availability[[]]){

WatchstanderAvailability[i,[]] = WatchstanderAvailability(M\_Availability[i,[])

}

Return WatchstanderAvailability[[]]

}

//Calculates total amount of shift points for each day of the month

Input = V\_month vector

Output = Monthly\_total []

1: Full work days

2: Days leading into a weekend

3: Weekend days

4: Final day of the weekend.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

// Takes the month vector and a String and then determines how many watch points the shift gets

Inputs:

Month vector

Shift string that should be “0”, “D”, or “N”

Integer for what day of the month it is

Output: Integer value of shift points

Input code:

1: Full work days

2: Days leading into a weekend

3: Weekend days

4: Final day of the weekend.

*Function* ShiftEvaluator (V\_month[], ShiftDorN, Date) ={

If(ShiftDorN == “D”){

If V\_month[Date] ==1:

Return 4\*WODH + 8\*WH

Else if V\_month[Date] ==2:

Return 8\*WODH + 4\*WEH

Else if V\_month[Date] == 3 or 4:

Return 12\*WEH

Else:

Print (invalid input given in input vector)

Return Null

}

}

Else if (ShiftDorN == “N”){

If V\_month[Date] ==1:

Return 12\*WODH

Else if V\_month[Date] == 2 or 3:

Return 12\*WEH

Else if V\_month[Date] == 4:

Return 4\*WEH + 8\*WODH

Else:

Print (invalid input given in input vector)

Return Null

}

}

Else if (ShiftDorN == 0) {

Return 0

}

Else{

Print (invalid input given in input vector)

Return Null

}

}

**Stochastic Model**

//Baseline model that will randomly assign watches as available. Intention is to update this with an updating probability vector based on number of watches already stood during the month.

Inputs: M\_Availability[[]], Monthly\_total[]

Output: Results[], Watchbill[[]]. It will be a tuple where the first output will be a vector of each persons watch points. This can be compared to the ideal vector to determine model validity. The Watchbill will be a matrix that describes when each person has watch.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//This first function will randomly generate a functional watchbill.

Input: Availability matrix

Output: A viable watchbill

*Function* BaselineWBGenerator (M\_Availability[[]]) {

//Defines Matrix that says if people are available to stand watch. Matrix code:

0: Not available for any watch

1: Available for Day watch only

2: Available for Night watch only

3: Available for either watch

//

OutputWatchbill = [[]]

OutputResults = []

WatchstanderAvailability = WatchstanderAvailability(M\_Availability[[]])

//Look at each day of the month

For j in num.col(WatchstanderAvailability[[]]){

PotentialWatchstanders = WatchstanderAvailability[[],j]

While (){

counter = 0

WatchstanderDay = randint(1, numrows(PotentialWatchstanders)

WatchstanderNight = randint(1, numrows(PotentialWatchstanders)

if(WatchstanderDay ==WatchstanderNight){

Counter +=1

continue

}

if(PotentialWatchstanders[WatchstanderDay] == 3 or 1 and PotentialWatchstanders[WatchstanderNight] ==3 or 2){

//Check if they had watch yesterday

YesterdayPers1 = OutputWatchbill[WatchstanderDay,j]

YesterdayPers2 = OutputWatchbill[WatchstanderNight,j]

If(YesterdayPers1 == “D” or “N” or YesterdayPers2 == “D” or “N”){

Counter +=1

continue

}

Else{

OutputWatchbill[WatchstanderDay,j] = “D”

OutputWatchbill[WatchstanderNight,j] = “N”

break

}

}

If (counter >50){

Print (Counter exceeds 50)

Return Null

}

}

}

Return OutputWatchbill[[]]

}

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

//Takes a completed watchbill and evaluates it for watch points

*Function* WatchbillEvaluator (OutputWatchbill[[]], Vmonth[]){

}