lista_5

July 19, 2018

MAI 103: Análise de Risco // Prof. Eber
Lista 05 // Data: 10/07/2018 // Entrega: 17/07/2018
Luis Filipe Kopp

```
In [1]: set.seed(1)
        library(triangle)
        library(igraph)
        library(hexbin)
        library(ggplot2)
```

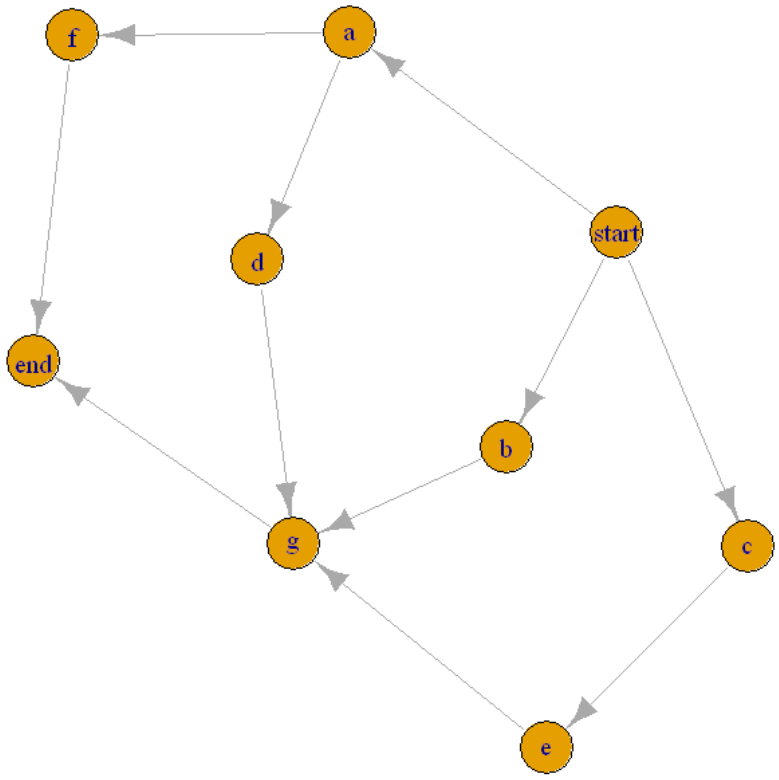
```
Warning message:
"package 'triangle' was built under R version 3.3.3"Warning message:
"package 'igraph' was built under R version 3.3.3"
Attaching package: 'igraph'

The following objects are masked from 'package:stats':

    decompose, spectrum

The following object is masked from 'package:base':

    union

Warning message:
"package 'hexbin' was built under R version 3.3.3"
```

1-Usando MC na rede descrita nas tabelas 1 e 2, obtenha:

ativ	Min	Mprov	Max
A	1	2	4
B	5	6	7
C	2	4	5
D	1	3	4
E	4	5	7
F	3	4	5
G	1	2	3

Tabela 1 - Atvidades
(start - A B C) (A - F D) (B - G) (C - E) (D - G) (E - G) (F - end) (G - end)
Tabela 2 - Redes

```
In [2]: g1 <- graph(edges=c('start','a', 'start','b', 'start', 'c','c','e','a','f',
                            'a','d','f','end','g','end','b','g','e','g','d','g') )
        plot(g1)
```



Uma aproximação empírica para a duração do projeto mostrado.

```
In [38]: resposta_final <- c()
         ant <- c()

         simula <- function(p){
```

```
        resposta <- c()

        n <- length(p)
        procura_caminho <- function(p,node,anterior,tempo){
            if(node == "end"){
                if(tempo > t){
                    print(anterior)
                    ant <<- anterior
                    t <<- tempo
                }

                resposta <<- c(resposta,tempo)

            }
            else{
                for(i in 1:n){
                    if(p[[i]][1] == node){
                        t <- tempo + as.numeric(p[[i]][3])
                        procura_caminho(p,p[[i]][2], paste(anterior,p[[i]][1],
                                                           sep = "-"), t)
                    }
                }
            }

        }
        procura_caminho(p,"start","",0)
        resposta_final <<- c(resposta_final,max(resposta))
        }
```

In [39]:
```
        resposta_final <- c()
        caminhos <- c()
        t <- 0
        precedencia <-  list(c("start","A",2),c("start","B",6),
                             c("start","C",4),c("A","F",4),
                             c("A","D",3),c("B","G",2),
                             c("C","E",5),c("D","G",2),
                             c("E","G",2),c("F","end",0),c("G","end",0))
        simula(precedencia)
        resposta_final
        ant
```

```
[1] "-start-A-F"
[1] "-start-A-D-G"
[1] "-start-B-G"
[1] "-start-C-E-G"
```

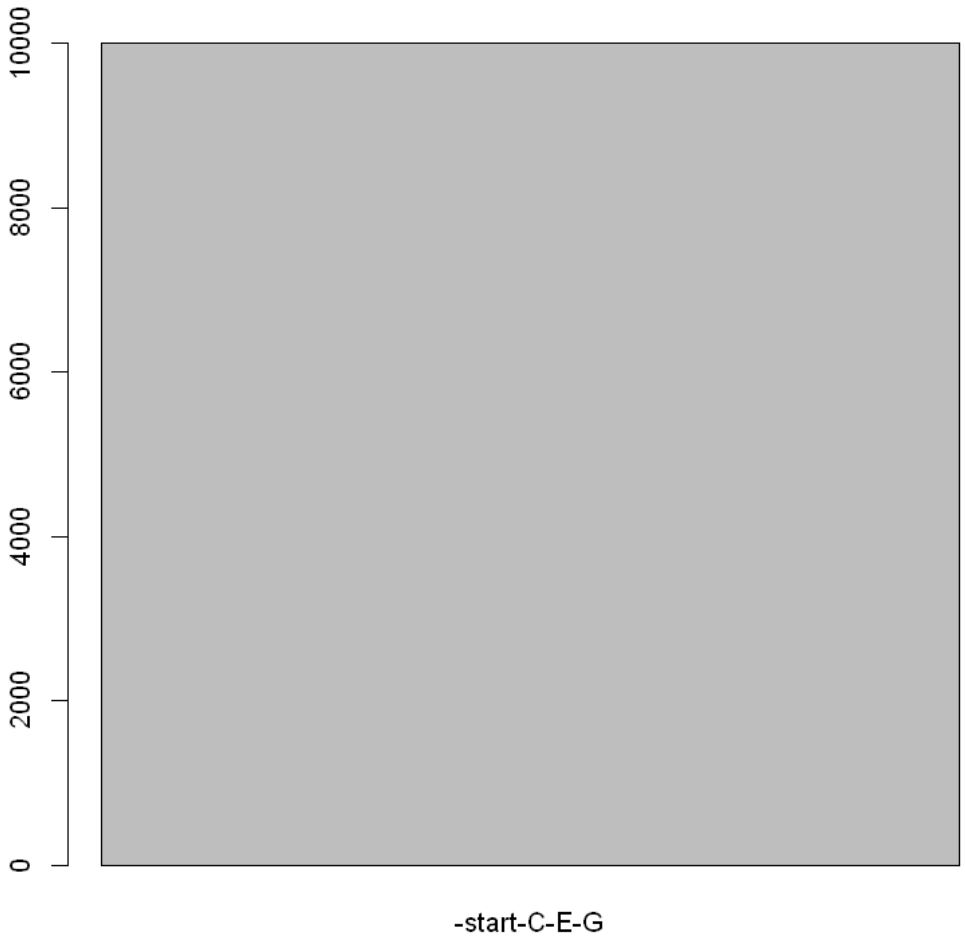
11

'-start-C-E-G'

Obtenha um estimativa das probabilidades das atividades pertencerem ao caminho crítico

```
In [40]: resposta_final <- c()
         caminho <- c()
         for( h in 1:10000){
         precedencia <-  list(c("start","A",rtriangle(1,1,4,2)),
                              c("start","B",rtriangle(1,5,7,6)),
                              c("start","C",rtriangle(1,2,5,4)),
                              c("A","F",rtriangle(1,3,5,4)),
                              c("A","D",rtriangle(1,1,4,3)),
                              c("B","G",rtriangle(1,1,3,2)),
                              c("C","E",rtriangle(1,4,7,5)),
                              c("D","G",rtriangle(1,1,3,2)),
                              c("E","G",rtriangle(1,1,3,2)),
                              c("F","end",0),c("G","end",0))
         simula(precedencia)
         caminho <<- c(caminho,ant)
         }
         barplot(table(caminho))
```

```
[1] "-start-C-E-G"
[1] "-start-C-E-G"
[1] "-start-C-E-G"
[1] "-start-C-E-G"
[1] "-start-C-E-G"
[1] "-start-C-E-G"
[1] "-start-C-E-G"
[1] "-start-C-E-G"
[1] "-start-C-E-G"
```



Uma análise empírica sugere que se B estiver perto do máximo (7) e C e E estiverem perto do mínimo (2 e 4), o caminho crítico passaria a ser start-B-G-end ao invés de start-C-E-G-end, porém rodamos a simulação n vezes (para n muito grande) e em mais de 99.9% não houve troca de caminho crítico, por isso podemos assumir como extremamente improvável.

```
In [6]: n <- 1000000
        c <- rtriangle(n,2,5,4) + rtriangle(n,4,7,5) - rtriangle(n,5,7,6)
        sum(c>=0)/length(c)
```

0.99968

Compare o resultado com aquele obtido pela aproximação PERT.

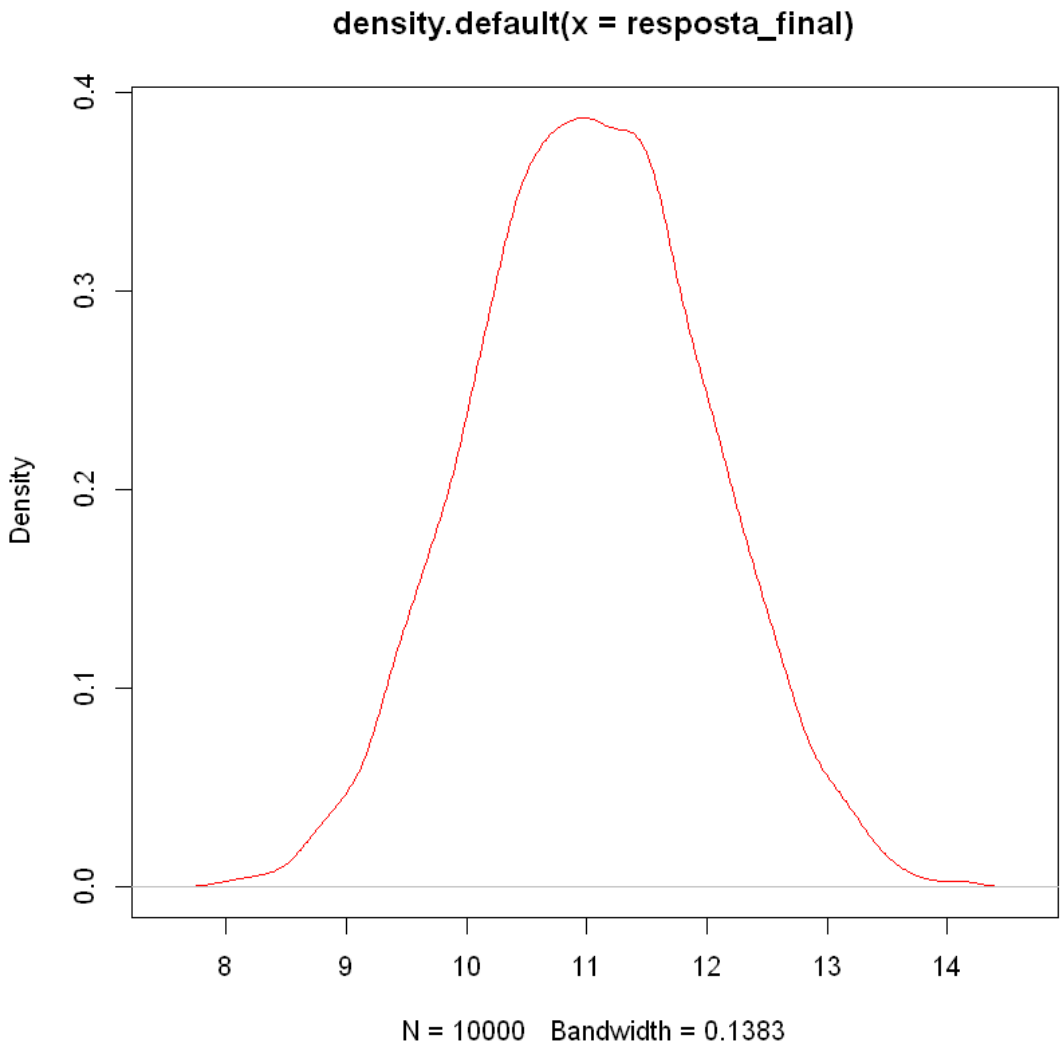
```
In [7]: resposta_final <- c()
        for( h in 1:10000){
```

```
precedencia <-  list(c("start","A",rtriangle(1,1,4,2)),
                     c("start","B",rtriangle(1,5,7,6)),
                     c("start","C",rtriangle(1,2,5,4)),
                     c("A","F",rtriangle(1,3,5,4)),
                     c("A","D",rtriangle(1,1,4,3)),
                     c("B","G",rtriangle(1,1,3,2)),
                     c("C","E",rtriangle(1,4,7,5)),
                     c("D","G",rtriangle(1,1,3,2)),
                     c("E","G",rtriangle(1,1,3,2)),
                     c("F","end",0),c("G","end",0))
simula(precedencia)
}
mean(resposta_final)
```

11.0263285705205

Em cerca de 50% das simulações, o tempo de conclusão foi superior à 11.

```
In [8]: plot(density(resposta_final), col="red")
```



Gere todos agendamentos possíveis para o minimo prazo.

```
In [55]: precedencia <-  list(c(1,2,2), c(1,3,6), c(1,4,4), c(2,7,4), c(2,5,3), c(3,8,2),
                              c(4,6,5), c(5,8,2), c(6,8,2), c(7,9,0),c(8,9,0))

         n <- 9
         z <- length(precedencia)
         inicio_cedo <- rep(0,n)

         for(i in 1:n){
             for(j in 1:z){
                 if( i == precedencia[[j]][2]){
                     item <- precedencia[[j]][1]
                     if(inicio_cedo[[i]] <= inicio_cedo[[item]] + precedencia[[item]][3]){
```

```
                    inicio_cedo[[i]] <- inicio_cedo[[item]] + precedencia[[item]][3]
                }
            }

        }
    }

    inicio_tarde <- rep(inicio_cedo[[n]],n)

    for(i in n:1){
        for(j in 1:z){
            if( i == precedencia[[j]][1]){
                item <- precedencia[[j]][2]
                if(inicio_tarde[[i]] >= inicio_tarde[[item]] - precedencia[[i]][3]){
                    inicio_tarde[[i]] <- inicio_tarde[[item]] - precedencia[[i]][3]
                }
            }

        }
    }

    inicio_cedo
    inicio_tarde
```

1. 0 2. 2 3. 2 4. 2 5. 8 6. 6 7. 8 8. 11 9. 13
1. 0 2. 2 3. 7 4. 5 5. 8 6. 9 7. 8 8. 11 9. 13

In [128]:
```
agendamento <- c()
varia <- function(inicio, cedo,tarde,pos){
    for(i in cedo[[pos]]:tarde[[pos]]){
        inicio[[pos]] <- i
        agendamento <<- rbind(agendamento,inicio)
    }


}
inicio <- inicio_cedo
varia(inicio, inicio_cedo, inicio_tarde,4)

n <- length(agendamento)/length(inicio)


for( g in 1:length(inicio)){

for(a in g:n){
 varia(agendamento[a,], inicio_cedo, inicio_tarde,a)

}
```

```
    }

agendamento
```

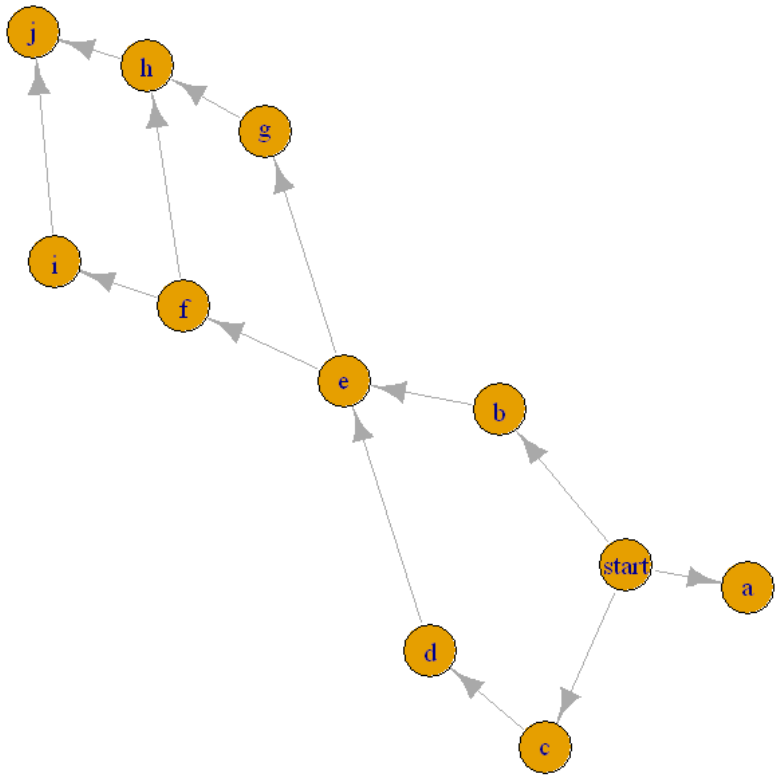
inicio	0	2	2	2	8	6	8	11	13
inicio	0	2	2	3	8	6	8	11	13
inicio	0	2	2	4	8	6	8	11	13
inicio	0	2	2	5	8	6	8	11	13
inicio	0	2	2	2	8	6	8	11	13
inicio	0	2	2	3	8	6	8	11	13
inicio	0	2	2	4	8	6	8	11	13
inicio	0	2	3	4	8	6	8	11	13
inicio	0	2	4	4	8	6	8	11	13
inicio	0	2	5	4	8	6	8	11	13
inicio	0	2	6	4	8	6	8	11	13
inicio	0	2	7	4	8	6	8	11	13
inicio	0	2	2	2	8	6	8	11	13
inicio	0	2	2	3	8	6	8	11	13
inicio	0	2	2	4	8	6	8	11	13
inicio	0	2	2	5	8	6	8	11	13
inicio	0	2	2	3	8	6	8	11	13
inicio	0	2	2	4	8	6	8	11	13
inicio	0	2	3	4	8	6	8	11	13
inicio	0	2	4	4	8	6	8	11	13
inicio	0	2	5	4	8	6	8	11	13
inicio	0	2	6	4	8	6	8	11	13
inicio	0	2	7	4	8	6	8	11	13
inicio	0	2	2	2	8	6	8	11	13
inicio	0	2	2	3	8	6	8	11	13
inicio	0	2	2	4	8	6	8	11	13
inicio	0	2	2	5	8	6	8	11	13
inicio	0	2	2	4	8	6	8	11	13
inicio	0	2	3	4	8	6	8	11	13
inicio	0	2	4	4	8	6	8	11	13
...	...	...	...	...	...	...	...	...	...
inicio	0	2	2	3	8	8	8	11	13
inicio	0	2	2	3	8	9	8	11	13
inicio	0	2	2	2	8	6	8	11	13
inicio	0	2	2	2	8	6	8	11	13
inicio	0	2	2	3	8	6	8	11	13
inicio	0	2	2	4	8	6	8	11	13
inicio	0	2	2	5	8	6	8	11	13
inicio	0	2	3	4	8	6	8	11	13
inicio	0	2	2	4	8	6	8	11	13
inicio	0	2	2	3	8	6	8	11	13
inicio	0	2	2	3	8	7	8	11	13
inicio	0	2	2	3	8	8	8	11	13
inicio	0	2	2	3	8	9	8	11	13
inicio	0	2	2	2	8	6	8	11	13
inicio	0	2	2	2	8	6	8	11	13
inicio	0	2	2	3	8	6	8	11	13
inicio	0	2	2	4	8	6	8	11	13
inicio	0	2	2	5	8	6	8	11	13
inicio	0	2	4	4	8	6	8	11	13
inicio	0	2	3	4	8	6	8	11	13
inicio	0	2	2	4	8	6	8	11	13
inicio	0	2	2	3	8	6	8	11	13

2- Observe os dados sobre o projeto de uma obra mostrado nas tabela 3. Calcule:

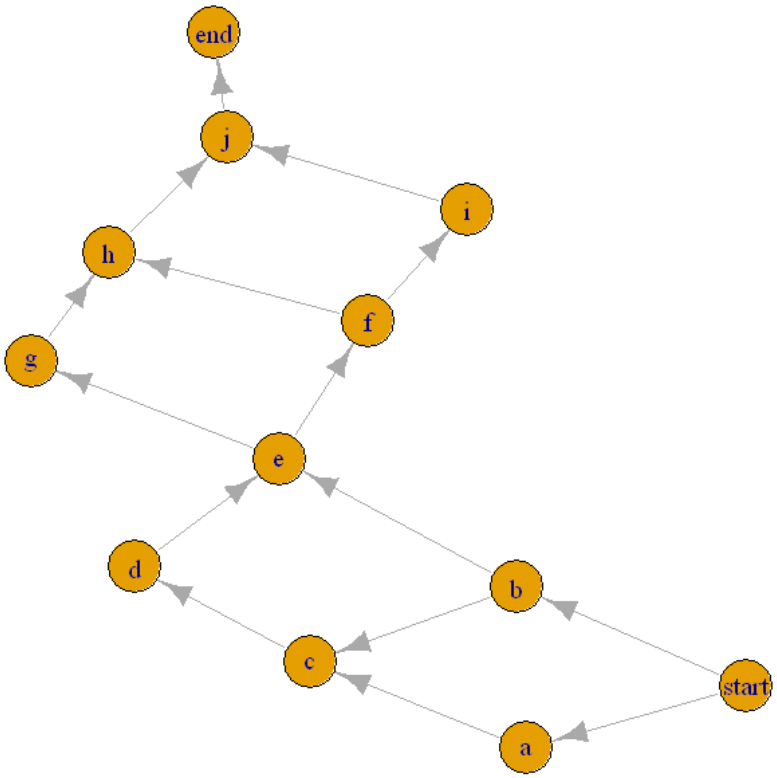
Atividade	Descrição	Pred	DMin	DMp	DMax	CMin	CMp	CMax
A	Obter materiais	START	2	4	18	300	450	600
B	Obter mão de obra	START	5	9	19	480	600	720
C	Escavar	START	4	10	28	3750	4500	5250
D	Colocar fundação	C	8	13	36	8400	9600	10800
E	Construir Estrutura	B,D	44	60	100	300000	312000	322500
F	Instalação Hidráulica	E	30	40	74	37650	39600	41400
G	Instalação Elétrica	E	9	20	43	10500	11550	12600
H	Acabamento Interior	F,G	24	30	48	36000	38400	40800
I	Acabamento Exterior	F	28	29	96	48750	52500	56250
J	Limpeza Local	H,I	10	10	12	360	450	540

Tabela 3 - Atividades

```
In [10]: g2 <- graph( edges=c('start','a', 'start','b', 'start', 'c','c','d','b','e',
                               'd','e','e','f','e','g','f','h','g','h','f','i','h',
                               'j','i','j') )
         plot(g2)
```



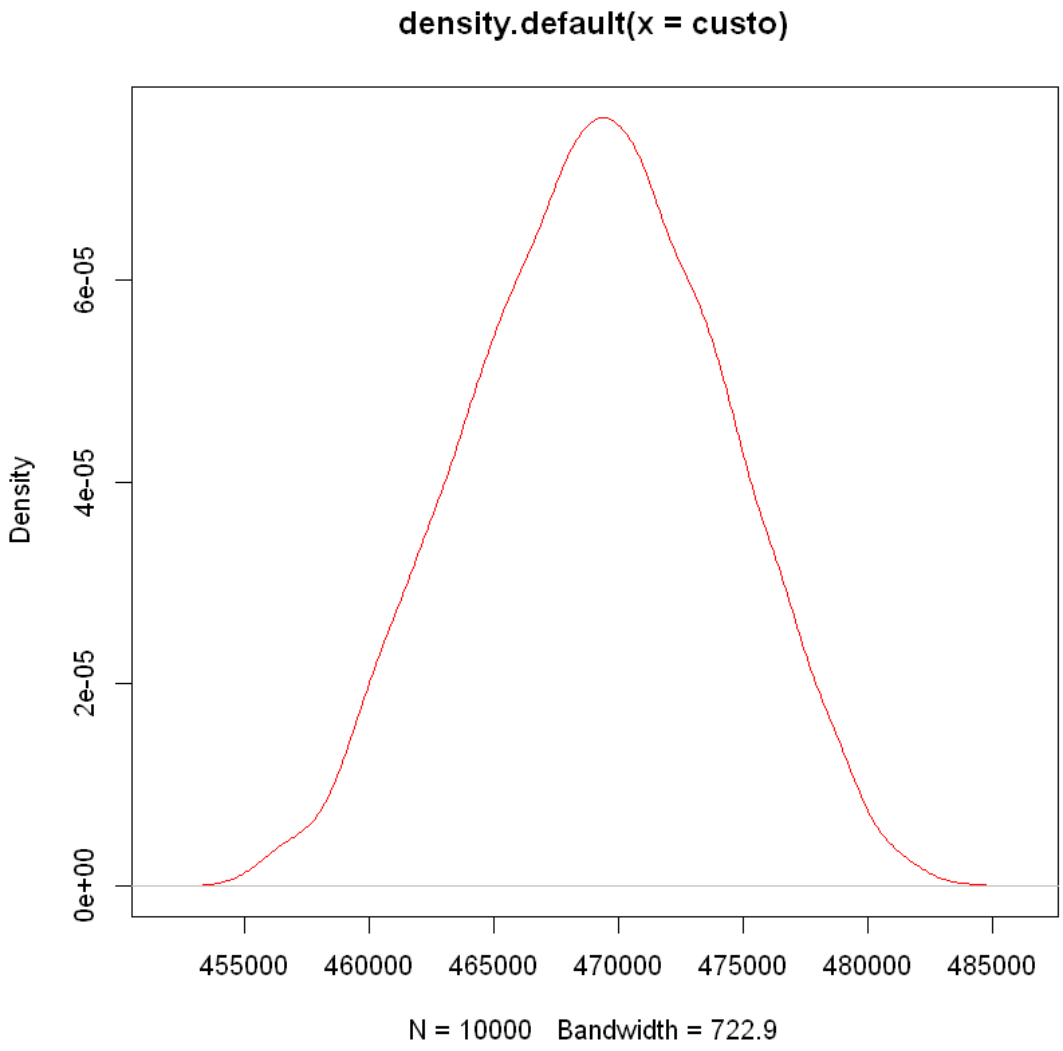

```
In [11]: g2 <- graph( edges=c('start','a', 'start','b', 'a', 'c','b','c','c','d',
                              'b','e','d','e','e','f','e','g','f','h','g','h','f',
                              'i','h','j','i','j','j','end') )
         plot(g2)
```



Obtenha as aproximações para o risco de custo e de prazo da obra

```
In [12]: custo <- (rtriangle(10000,300,600,450)+
                   rtriangle(10000,480,720,600)+
                   rtriangle(10000,3750,5250,4500)+
                   rtriangle(10000,8400,10800,9600)+
                   rtriangle(10000,300000,322500,312000)+
                   rtriangle(10000,37650,41400,39600)+
                   rtriangle(10000,10500,12600,11550)+
                   rtriangle(10000,36000,40800,38400)+
                   rtriangle(10000,48750,56250,52500)+
                   rtriangle(10000,360,540,450))
         mean(custo)
         plot(density(custo), col="red")
```

469074.644877677



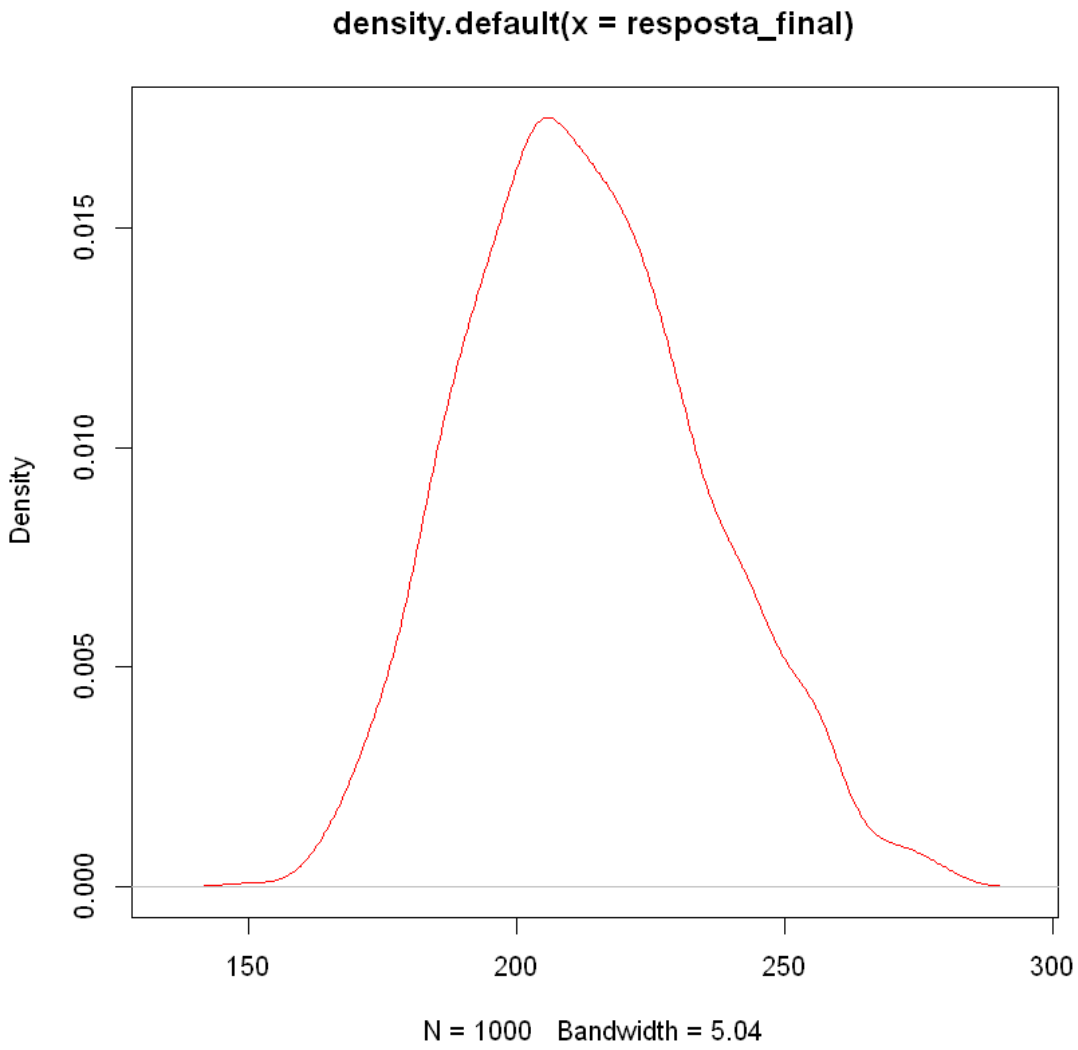

```
In [13]: resposta_final <- c()
         for( h in 1:1000){
         precedencia <-  list(c("start","A",rtriangle(1,2,18,4)),
                              c("start","B",rtriangle(1,5,19,9)),
                              c("start","C",rtriangle(1,4,28,10)),
                              c("C","D",rtriangle(1,8,36,13)),
                              c("B","E",rtriangle(1,44,100,60)),
                              c("D","E",rtriangle(1,44,100,60)),
                              c("E","F",rtriangle(1,30,74,40)),
                              c("E","G",rtriangle(1,9,43,20)),
                              c("F","H",rtriangle(1,24,48,30)),
```

```
                                c("G","H",rtriangle(1,24,48,30)),
                                c("F","I",rtriangle(1,28,96,29)),
                                c("H","J",rtriangle(1,10,12,10)),
                                c("I","J",rtriangle(1,10,12,10)),
                                c("J","end",0),
                                c("J","end",0))
        simula(precedencia)
        }
        mean(resposta_final)
```

```
[1] "-start-B-E-F-H-J"
[1] "-start-B-E-F-I-J"
[1] "-start-C-D-E-F-I-J"
[1] "-start-B-E-F-I-J"
[1] "-start-C-D-E-F-I-J"
[1] "-start-C-D-E-F-I-J"
[1] "-start-C-D-E-F-I-J"
[1] "-start-C-D-E-F-I-J"
[1] "-start-C-D-E-F-I-J"
```

213.00974275867

In [14]:
```
plot(density(resposta_final), col="red")
```




```
In [15]: resposta_final <- c()
         for( h in 1:1000){
         precedencia <-  list(c("start","A",rtriangle(1,2,18,4)),
                              c("start","B",rtriangle(1,5,19,9)),
                              c("B","C",rtriangle(1,4,28,10)),
                              c("A","C",rtriangle(1,4,28,10)),
                              c("C","D",rtriangle(1,8,36,13)),
                              c("B","E",rtriangle(1,44,100,60)),
                              c("D","E",rtriangle(1,44,100,60)),
                              c("E","F",rtriangle(1,30,74,40)),
                              c("E","G",rtriangle(1,9,43,20)),
                              c("F","H",rtriangle(1,24,48,30)),
                              c("G","H",rtriangle(1,24,48,30)),
```

```
                                c("F","I",rtriangle(1,28,96,29)),
                                c("H","J",rtriangle(1,10,12,10)),
                                c("I","J",rtriangle(1,10,12,10)),
                                c("J","end",0),
                                c("J","end",0))
        simula(precedencia)
        }
        mean(resposta_final)
```

```
[1] "-start-B-C-D-E-F-I-J"
[1] "-start-B-C-D-E-F-I-J"
[1] "-start-A-C-D-E-F-I-J"
[1] "-start-B-C-D-E-F-I-J"
[1] "-start-B-C-D-E-F-I-J"
```

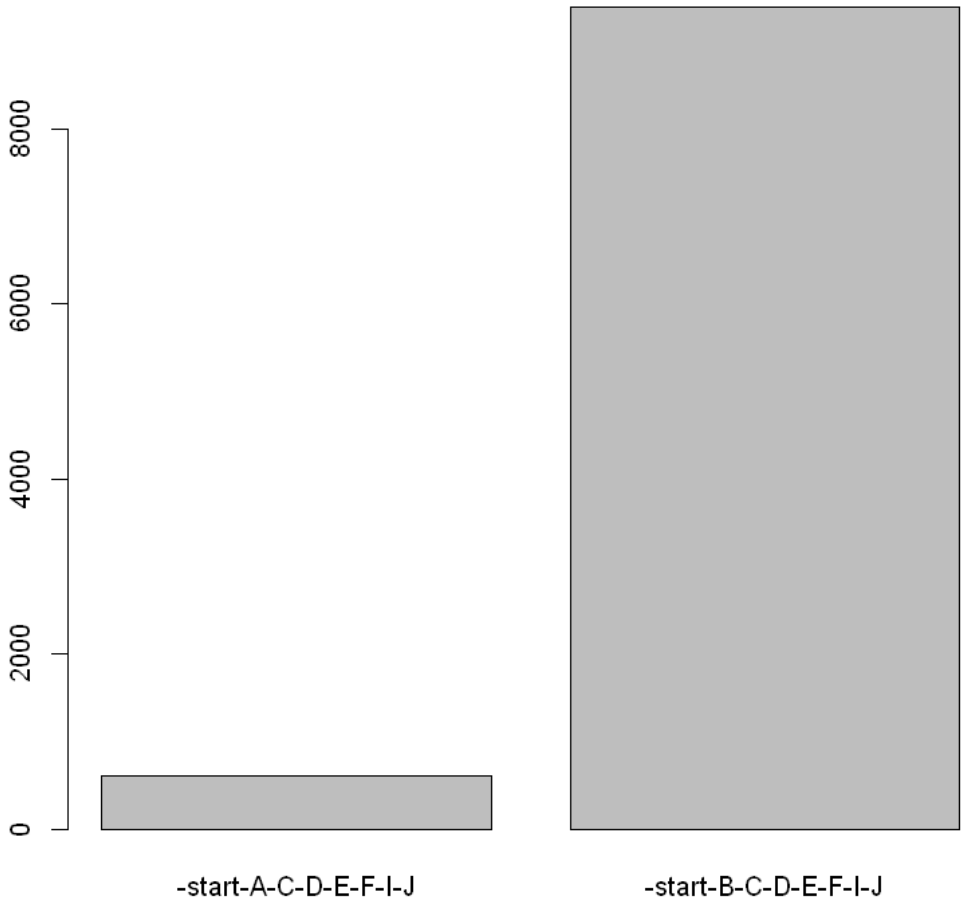
224.77359454236

Obtenha uma estimativa das probabilidades das atividades pertencerem ao caminho crítico

```
In [16]: resposta_final <- c()
         caminho <- c()
         for( h in 1:10000){
         precedencia <-  list(c("start","A",rtriangle(1,2,18,4)),
                              c("start","B",rtriangle(1,5,19,9)),
                              c("B","C",rtriangle(1,4,28,10)),
                              c("A","C",rtriangle(1,4,28,10)),
                              c("C","D",rtriangle(1,8,36,13)),
                              c("B","E",rtriangle(1,44,100,60)),
                              c("D","E",rtriangle(1,44,100,60)),
                              c("E","F",rtriangle(1,30,74,40)),
                              c("E","G",rtriangle(1,9,43,20)),
                              c("F","H",rtriangle(1,24,48,30)),
                              c("G","H",rtriangle(1,24,48,30)),
                              c("F","I",rtriangle(1,28,96,29)),
                              c("H","J",rtriangle(1,10,12,10)),
                              c("I","J",rtriangle(1,10,12,10)),
                              c("J","end",0),
                              c("J","end",0))
         simula(precedencia)
         caminho <<- c(caminho,ant)
         }
         mean(resposta_final)
         barplot(table(caminho))
```

```
[1] "-start-A-C-D-E-F-I-J"
```

224.807260112853



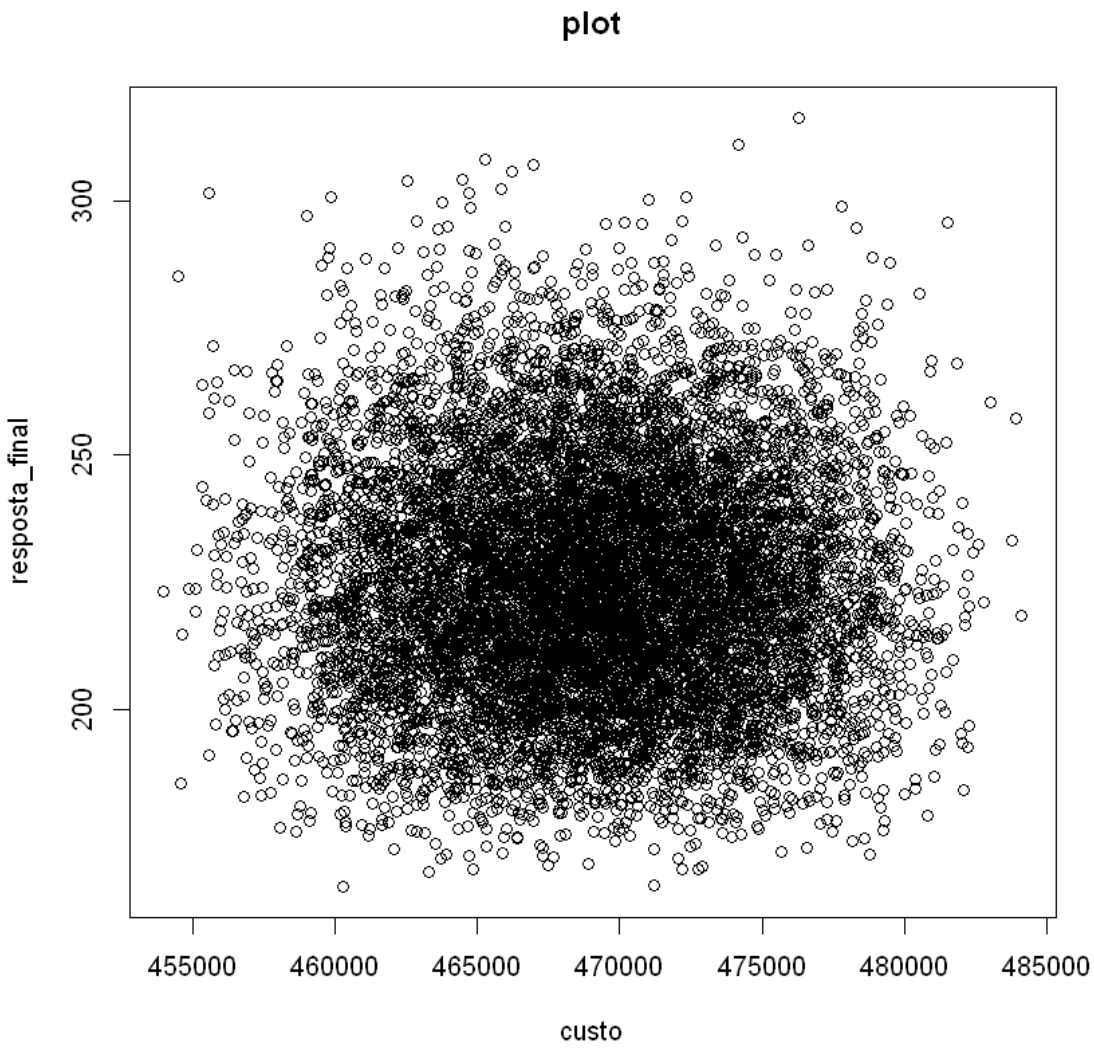
Faça um grafico de dispersão mostrando a correlação entre prazo e custo da obra.

```
In [17]: cor(custo,resposta_final)
```

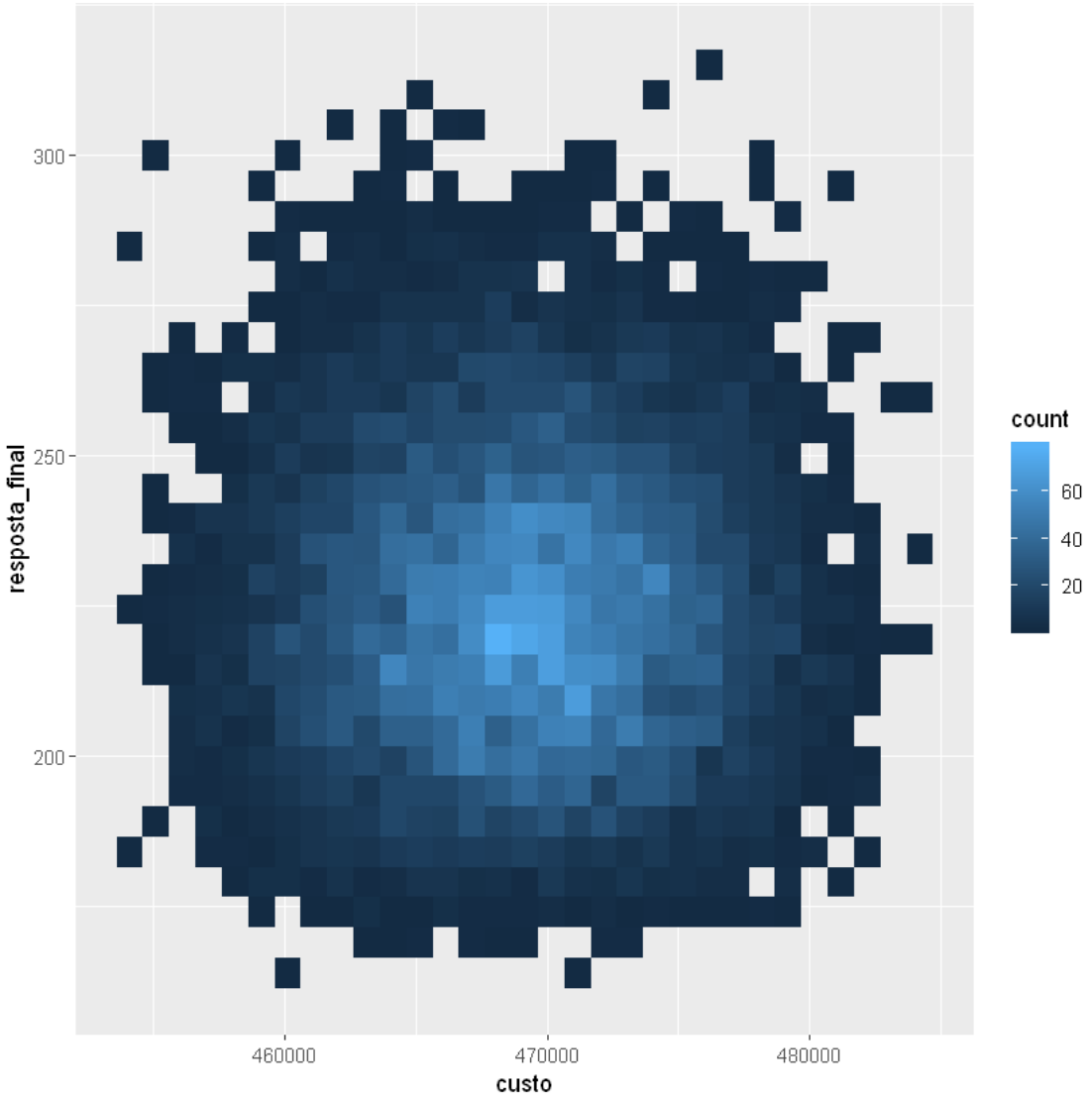
-0.00639471054887277

A correlação entre custo e prazo é praticamente nula, indicando que, neste modelo, uma variação no prazo não contribui para a variação do custo.

```
In [18]: plot(custo,resposta_final, main="plot")
```




```
In [19]: df <- data.frame(custo,resposta_final)
         qplot(custo,resposta_final,data=df, geom='bin2d')
```

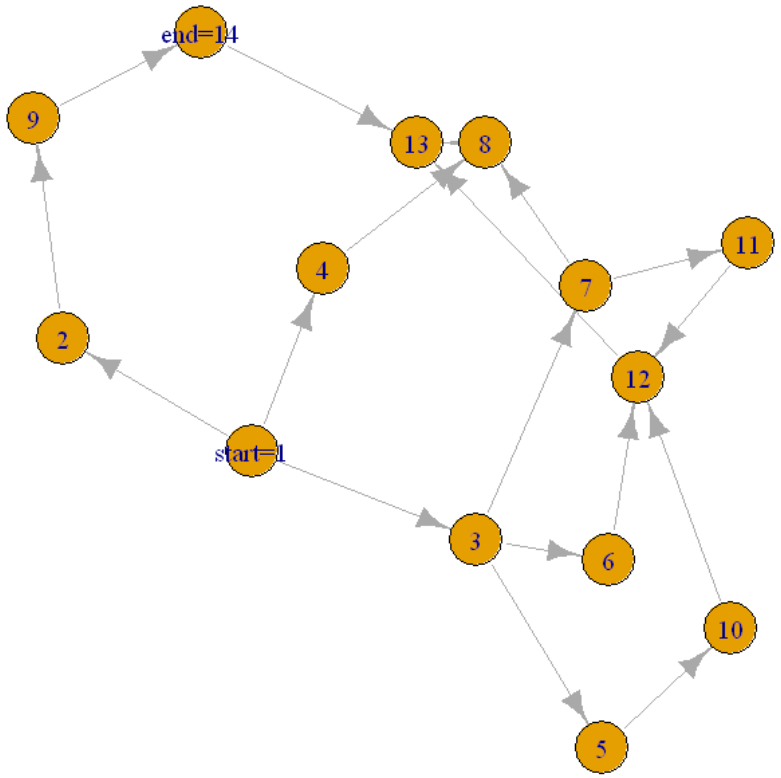


3- Dado a rede de projeto mostrada na figura 1, aplique a técnica de MC para obter um agendamento que proporciona uma boa aproximação para o máximo NPV a ser obtido pelo projeto. O número na parte de cima do circulo representando uma atividade é sua duração e o abaixo é o seu NPV trazido para o término da atividade. Assuma:
que a data limite n = 44
valor presente é continuamente descontado a uma taxa r = 0.01
Passos:
gerar a relação (atividade, est, lst)
implementar uma função que sorteia um cronograma
implementar uma função que verifica se o cronograma é valido
avaliar a qualidade da aproximação para a solução ótima

Atividade	Predecessor	Duração	NPV
1	START	0	0

Atividade	Predecessor	Duração	NPV
2	1	6	-140
3	1	5	318
4	1	3	312
5	3	1	-329
6	3	6	153
7	3	2	193
8	4,7	1	361
9	2	4	24
10	5	3	33
11	7	2	387
12	6,10,11	3	-386
13	8,12	5	171
14	9,13	0	0

```
In [20]: g3 <- graph(edges=c('start=1',2,'start=1',3,'start=1',4,3,5,3,6,3,7,
                              4,8,7,8,2,9,5,10,7,11,
                              6,12,10,12,11,12,8,13,12,13,'end=14',13,9,'end=14') )
         plot(g3)
```




```
In [21]: n <- 14
         duracao_max <- 44
         duracao <- c(0,6,5,3,1,6,2,1,4,3,2,3,5,0)
         valor <- c(0,-140,318,312,-329,153,193,361,24,33,387,-386,171,0)
         precedencia <- list(c(1,2),c(1,3),c(1,4),c(2,9),c(3,5),c(3,6),c(3,7),
                             c(4,8),c(5,10),c(6,12),c(7,8),c(7,11),c(8,13),
                             c(9,14),c(10,12),c(11,12),c(12,13),c(13,14))
         z <- length(precedencia)
         inicio_cedo <- rep(0,14)
         inicio_tarde <- rep(duracao_max,14)
         for(i in 1:n){
             for(j in 1:z){
                 if( i == precedencia[[j]][2]){
```

```
                item <- precedencia[[j]][1]
                if(inicio_cedo[[i]] <= inicio_cedo[[item]] + duracao[[item]]){
                    inicio_cedo[[i]] <- inicio_cedo[[item]] + duracao[[item]]
                }
            }

        }
    }

    for(i in n:1){
        for(j in 1:z){
            if( i == precedencia[[j]][1]){
                item <- precedencia[[j]][2]
                if(inicio_tarde[[i]] >= inicio_tarde[[item]] - duracao[[i]]){
                    inicio_tarde[[i]] <- inicio_tarde[[item]] - duracao[[i]]
                }
            }

        }
    }

    inicio_cedo
    inicio_tarde
```

1. 0 2. 0 3. 0 4. 0 5. 5 6. 5 7. 5 8. 7 9. 6 10. 6 11. 7 12. 11 13. 14 14. 19
1. 25 2. 34 3. 25 4. 35 5. 32 6. 30 7. 32 8. 38 9. 40 10. 33 11. 34 12. 36 13. 39 14. 44

In [22]:
```
valida <- function(inicio, precedencia, duracao, prazo){
    z <- length(precedencia)
    n <- length(inicio)
    for(i in 1:n){
        if(inicio[[i]] + duracao[[i]] > prazo) {
            return(0)
        }
    }

    for(i in 1:z){
        if(inicio[[precedencia[[i]][2]]] < inicio[[precedencia[[i]][1]]] +
           duracao[[precedencia[[i]][1]]]){
            return(0)
        }
    }
    return(1)
}
```

In [23]:
```
npv <- function(inicio, valor,taxa){
    npv0 <- 0
    n <- length(inicio)
```

```
        for(i in 1:n){
            npv0 <- npv0 + valor[[i]] * (1 - taxa)^(inicio[[i]] + duracao[[i]])
        }
        npv0
    }
```

In [24]:
```
adia <- function(inicio,duracao,precedencia){
    n <- length(inicio)
    z <- length(precedencia)
    for(j in 1:z){
        if( inicio[[precedencia[[j]][1]]] +
           duracao[[precedencia[[j]][1]]] >  inicio[[precedencia[[j]][2]]]){
            inicio[[precedencia[[j]][2]]] <- inicio[[precedencia[[j]][1]]] +
            duracao[[precedencia[[j]][1]]]

            if(valida(inicio, precedencia, duracao, duracao_max)){

                return(inicio)
            }
        }
    }
    if(valida(inicio, precedencia, duracao, duracao_max)){
        return(inicio)
    }
}
```

In [25]:
```
inicio <- inicio_cedo
for(i in 1:n){
    novo <- inicio
    v <- inicio_tarde[[i]] - novo[[i]]
    for(u in novo[[i]]:inicio_tarde[[i]]){
    novo[[i]] <- u
    novo <- adia(novo,duracao,precedencia)
                if(valida(novo, precedencia, duracao, duracao_max)){
            print(c(round(novo), round(npv(novo, valor,0.01))))
        }
    if(npv(novo, valor,0.01) > npv(inicio, valor,0.01) ){
        if(valida(novo, precedencia, duracao, duracao_max)){
            inicio <- novo
        }
    }
}
    }
inicio
npv(inicio, valor,0.01)
```

```
 [1]    0    0    0    0    5    5    5    7    6    6    7   11   14   19 1025
 [1]    1    1    1    1    6    6    6    8    7    7    8   12   15   20 1015
```

```
[1]    2    2    2    2    7    7    7    9    8    8    9   13   16   21 1005
[1]    3    3    3    3    8    8    8   10    9    9   10   14   17   22 994
[1]    4    4    4    4    9    9    9   11   10   10   11   15   18   23 985
[1]    5    5    5    5   10   10   10   12   11   11   12   16   19   24 975
[1]    6    6    6    6   11   11   11   13   12   12   13   17   20   25 965
[1]    7    7    7    7   12   12   12   14   13   13   14   18   21   26 955
[1]    8    8    8    8   13   13   13   15   14   14   15   19   22   27 946
[1]    9    9    9    9   14   14   14   16   15   15   16   20   23   28 936
[1]   10   10   10   10   15   15   15   17   16   16   17   21   24   29 927
[1]   11   11   11   11   16   16   16   18   17   17   18   22   25   30 918
[1]   12   12   12   12   17   17   17   19   18   18   19   23   26   31 908
[1]   13   13   13   13   18   18   18   20   19   19   20   24   27   32 899
[1]   14   14   14   14   19   19   19   21   20   20   21   25   28   33 890
[1]   15   15   15   15   20   20   20   22   21   21   22   26   29   34 881
[1]   16   16   16   16   21   21   21   23   22   22   23   27   30   35 873
[1]   17   17   17   17   22   22   22   24   23   23   24   28   31   36 864
[1]   18   18   18   18   23   23   23   25   24   24   25   29   32   37 855
[1]   19   19   19   19   24   24   24   26   25   25   26   30   33   38 847
[1]   20   20   20   20   25   25   25   27   26   26   27   31   34   39 838
[1]   21   21   21   21   26   26   26   28   27   27   28   32   35   40 830
[1]   22   22   22   22   27   27   27   29   28   28   29   33   36   41 822
[1]   23   23   23   23   28   28   28   30   29   29   30   34   37   42 813
[1]   24   24   24   24   29   29   29   31   30   30   31   35   38   43 805
[1]   25   25   25   25   30   30   30   32   31   31   32   36   39   44 797
[1]    0    0    0    0    5    5    5    7    6    6    7   11   14   19 1025
[1]    0    1    0    0    5    5    5    7    7    6    7   11   14   19 1026
[1]    0    2    0    0    5    5    5    7    8    6    7   11   14   19 1027
[1]    0    3    0    0    5    5    5    7    9    6    7   11   14   19 1028
[1]    0    4    0    0    5    5    5    7   10    6    7   11   14   19 1029
[1]    0    5    0    0    5    5    5    7   11    6    7   11   14   19 1030
[1]    0    6    0    0    5    5    5    7   12    6    7   11   14   19 1031
[1]    0    7    0    0    5    5    5    7   13    6    7   11   14   19 1032
[1]    0    8    0    0    5    5    5    7   14    6    7   11   14   19 1033
[1]    0    9    0    0    5    5    5    7   15    6    7   11   14   19 1034
[1]    0   10    0    0    5    5    5    7   16    6    7   11   14   20 1035
[1]    0   11    0    0    5    5    5    7   17    6    7   11   14   21 1036
[1]    0   12    0    0    5    5    5    7   18    6    7   11   14   22 1037
[1]    0   13    0    0    5    5    5    7   19    6    7   11   14   23 1038
[1]    0   14    0    0    5    5    5    7   20    6    7   11   14   24 1039
[1]    0   15    0    0    5    5    5    7   21    6    7   11   14   25 1040
[1]    0   16    0    0    5    5    5    7   22    6    7   11   14   26 1041
[1]    0   17    0    0    5    5    5    7   23    6    7   11   14   27 1042
[1]    0   18    0    0    5    5    5    7   24    6    7   11   14   28 1043
[1]    0   19    0    0    5    5    5    7   25    6    7   11   14   29 1044
[1]    0   20    0    0    5    5    5    7   26    6    7   11   14   30 1045
[1]    0   21    0    0    5    5    5    7   27    6    7   11   14   31 1046
[1]    0   22    0    0    5    5    5    7   28    6    7   11   14   32 1047
[1]    0   23    0    0    5    5    5    7   29    6    7   11   14   33 1048
```

```
[1]  0  24  0  0  5  5  5  7  30  6  7  11  14  34 1048
[1]  0  25  0  0  5  5  5  7  31  6  7  11  14  35 1049
[1]  0  26  0  0  5  5  5  7  32  6  7  11  14  36 1050
[1]  0  27  0  0  5  5  5  7  33  6  7  11  14  37 1051
[1]  0  28  0  0  5  5  5  7  34  6  7  11  14  38 1052
[1]  0  29  0  0  5  5  5  7  35  6  7  11  14  39 1053
[1]  0  30  0  0  5  5  5  7  36  6  7  11  14  40 1054
[1]  0  31  0  0  5  5  5  7  37  6  7  11  14  41 1054
[1]  0  32  0  0  5  5  5  7  38  6  7  11  14  42 1055
[1]  0  33  0  0  5  5  5  7  39  6  7  11  14  43 1056
[1]  0  34  0  0  5  5  5  7  40  6  7  11  14  44 1057
[1]  0  34  0  0  5  5  5  7  40  6  7  11  14  44 1057
[1]  0  34  1  0  6  6  6  8  40  7  8  12  15  44 1048
[1]  0  34  2  0  7  7  7  9  40  8  9  13  16  44 1040
[1]  0  34  3  0  8  8  8  10  40  9  10  14  17  44 1032
[1]  0  34  4  0  9  9  9  11  40  10  11  15  18  44 1024
[1]  0  34  5  0  10  10  10  12  40  11  12  16  19  44 1016
[1]  0  34  6  0  11  11  11  13  40  12  13  17  20  44 1008
[1]  0  34  7  0  12  12  12  14  40  13  14  18  21  44 1000
[1]  0  34  8  0  13  13  13  15  40  14  15  19  22  44 992
[1]  0  34  9  0  14  14  14  16  40  15  16  20  23  44 985
[1]  0  34  10  0  15  15  15  17  40  16  17  21  24  44 977
[1]  0  34  11  0  16  16  16  18  40  17  18  22  25  44 970
[1]  0  34  12  0  17  17  17  19  40  18  19  23  26  44 962
[1]  0  34  13  0  18  18  18  20  40  19  20  24  27  44 955
[1]  0  34  14  0  19  19  19  21  40  20  21  25  28  44 948
[1]  0  34  15  0  20  20  20  22  40  21  22  26  29  44 940
[1]  0  34  16  0  21  21  21  23  40  22  23  27  30  44 933
[1]  0  34  17  0  22  22  22  24  40  23  24  28  31  44 926
[1]  0  34  18  0  23  23  23  25  40  24  25  29  32  44 919
[1]  0  34  19  0  24  24  24  26  40  25  26  30  33  44 912
[1]  0  34  20  0  25  25  25  27  40  26  27  31  34  44 905
[1]  0  34  21  0  26  26  26  28  40  27  28  32  35  44 898
[1]  0  34  22  0  27  27  27  29  40  28  29  33  36  44 892
[1]  0  34  23  0  28  28  28  30  40  29  30  34  37  44 885
[1]  0  34  24  0  29  29  29  31  40  30  31  35  38  44 878
[1]  0  34  25  0  30  30  30  32  40  31  32  36  39  44 872
[1]  0  34  0  0  5  5  5  7  40  6  7  11  14  44 1057
[1]  0  34  0  1  5  5  5  7  40  6  7  11  14  44 1054
[1]  0  34  0  2  5  5  5  7  40  6  7  11  14  44 1051
[1]  0  34  0  3  5  5  5  7  40  6  7  11  14  44 1048
[1]  0  34  0  4  5  5  5  7  40  6  7  11  14  44 1045
[1]  0  34  0  5  5  5  5  8  40  6  7  11  14  44 1039
[1]  0  34  0  6  5  5  5  9  40  6  7  11  14  44 1032
[1]  0  34  0  7  5  5  5  10  40  6  7  11  14  44 1026
[1]  0  34  0  8  5  5  5  11  40  6  7  11  14  44 1020
[1]  0  34  0  9  5  5  5  12  40  6  7  11  14  44 1014
[1]  0  34  0  10  5  5  5  13  40  6  7  11  14  44 1008
```

```
[1]  0  34  0  11  5  5  5  14  40  6  7  11  15  44 1001
[1]  0  34  0  12  5  5  5  15  40  6  7  11  16  44 994
[1]  0  34  0  13  5  5  5  16  40  6  7  11  17  44 987
[1]  0  34  0  14  5  5  5  17  40  6  7  11  18  44 980
[1]  0  34  0  15  5  5  5  18  40  6  7  11  19  44 973
[1]  0  34  0  16  5  5  5  19  40  6  7  11  20  44 966
[1]  0  34  0  17  5  5  5  20  40  6  7  11  21  44 959
[1]  0  34  0  18  5  5  5  21  40  6  7  11  22  44 952
[1]  0  34  0  19  5  5  5  22  40  6  7  11  23  44 945
[1]  0  34  0  20  5  5  5  23  40  6  7  11  24  44 939
[1]  0  34  0  21  5  5  5  24  40  6  7  11  25  44 932
[1]  0  34  0  22  5  5  5  25  40  6  7  11  26  44 926
[1]  0  34  0  23  5  5  5  26  40  6  7  11  27  44 919
[1]  0  34  0  24  5  5  5  27  40  6  7  11  28  44 913
[1]  0  34  0  25  5  5  5  28  40  6  7  11  29  44 906
[1]  0  34  0  26  5  5  5  29  40  6  7  11  30  44 900
[1]  0  34  0  27  5  5  5  30  40  6  7  11  31  44 894
[1]  0  34  0  28  5  5  5  31  40  6  7  11  32  44 888
[1]  0  34  0  29  5  5  5  32  40  6  7  11  33  44 882
[1]  0  34  0  30  5  5  5  33  40  6  7  11  34  44 876
[1]  0  34  0  31  5  5  5  34  40  6  7  11  35  44 870
[1]  0  34  0  32  5  5  5  35  40  6  7  11  36  44 864
[1]  0  34  0  33  5  5  5  36  40  6  7  11  37  44 858
[1]  0  34  0  34  5  5  5  37  40  6  7  11  38  44 852
[1]  0  34  0  35  5  5  5  38  40  6  7  11  39  44 846
[1]  0  34  0  0  5  5  5  7  40  6  7  11  14  44 1057
[1]  0  34  0  0  6  5  5  7  40  7  7  11  14  44 1060
[1]  0  34  0  0  7  5  5  7  40  8  7  11  14  44 1062
[1]  0  34  0  0  8  5  5  7  40  9  7  12  15  44 1067
[1]  0  34  0  0  9  5  5  7  40  10  7  13  16  44 1072
[1]  0  34  0  0  10  5  5  7  40  11  7  14  17  44 1076
[1]  0  34  0  0  11  5  5  7  40  12  7  15  18  44 1081
[1]  0  34  0  0  12  5  5  7  40  13  7  16  19  44 1085
[1]  0  34  0  0  13  5  5  7  40  14  7  17  20  44 1090
[1]  0  34  0  0  14  5  5  7  40  15  7  18  21  44 1094
[1]  0  34  0  0  15  5  5  7  40  16  7  19  22  44 1098
[1]  0  34  0  0  16  5  5  7  40  17  7  20  23  44 1103
[1]  0  34  0  0  17  5  5  7  40  18  7  21  24  44 1107
[1]  0  34  0  0  18  5  5  7  40  19  7  22  25  44 1111
[1]  0  34  0  0  19  5  5  7  40  20  7  23  26  44 1116
[1]  0  34  0  0  20  5  5  7  40  21  7  24  27  44 1120
[1]  0  34  0  0  21  5  5  7  40  22  7  25  28  44 1124
[1]  0  34  0  0  22  5  5  7  40  23  7  26  29  44 1128
[1]  0  34  0  0  23  5  5  7  40  24  7  27  30  44 1132
[1]  0  34  0  0  24  5  5  7  40  25  7  28  31  44 1136
[1]  0  34  0  0  25  5  5  7  40  26  7  29  32  44 1140
[1]  0  34  0  0  26  5  5  7  40  27  7  30  33  44 1144
[1]  0  34  0  0  27  5  5  7  40  28  7  31  34  44 1148
```

```
[1]  0  34  0  0  28  5  5  7  40  29  7  32  35  44 1151
[1]  0  34  0  0  29  5  5  7  40  30  7  33  36  44 1155
[1]  0  34  0  0  30  5  5  7  40  31  7  34  37  44 1159
[1]  0  34  0  0  31  5  5  7  40  32  7  35  38  44 1163
[1]  0  34  0  0  32  5  5  7  40  33  7  36  39  44 1166
[1]  0  34  0  0  32  5  5  7  40  33  7  36  39  44 1166
[1]  0  34  0  0  32  6  5  7  40  33  7  36  39  44 1165
[1]  0  34  0  0  32  7  5  7  40  33  7  36  39  44 1164
[1]  0  34  0  0  32  8  5  7  40  33  7  36  39  44 1162
[1]  0  34  0  0  32  9  5  7  40  33  7  36  39  44 1161
[1]  0  34  0  0  32  10  5  7  40  33  7  36  39  44 1160
[1]  0  34  0  0  32  11  5  7  40  33  7  36  39  44 1158
[1]  0  34  0  0  32  12  5  7  40  33  7  36  39  44 1157
[1]  0  34  0  0  32  13  5  7  40  33  7  36  39  44 1156
[1]  0  34  0  0  32  14  5  7  40  33  7  36  39  44 1154
[1]  0  34  0  0  32  15  5  7  40  33  7  36  39  44 1153
[1]  0  34  0  0  32  16  5  7  40  33  7  36  39  44 1152
[1]  0  34  0  0  32  17  5  7  40  33  7  36  39  44 1151
[1]  0  34  0  0  32  18  5  7  40  33  7  36  39  44 1150
[1]  0  34  0  0  32  19  5  7  40  33  7  36  39  44 1148
[1]  0  34  0  0  32  20  5  7  40  33  7  36  39  44 1147
[1]  0  34  0  0  32  21  5  7  40  33  7  36  39  44 1146
[1]  0  34  0  0  32  22  5  7  40  33  7  36  39  44 1145
[1]  0  34  0  0  32  23  5  7  40  33  7  36  39  44 1144
[1]  0  34  0  0  32  24  5  7  40  33  7  36  39  44 1143
[1]  0  34  0  0  32  25  5  7  40  33  7  36  39  44 1141
[1]  0  34  0  0  32  26  5  7  40  33  7  36  39  44 1140
[1]  0  34  0  0  32  27  5  7  40  33  7  36  39  44 1139
[1]  0  34  0  0  32  28  5  7  40  33  7  36  39  44 1138
[1]  0  34  0  0  32  29  5  7  40  33  7  36  39  44 1137
[1]  0  34  0  0  32  30  5  7  40  33  7  36  39  44 1136
[1]  0  34  0  0  32  5  5  7  40  33  7  36  39  44 1166
[1]  0  34  0  0  32  5  6  8  40  33  8  36  39  44 1158
[1]  0  34  0  0  32  5  7  9  40  33  9  36  39  44 1149
[1]  0  34  0  0  32  5  8  10  40  33  10  36  39  44 1141
[1]  0  34  0  0  32  5  9  11  40  33  11  36  39  44 1132
[1]  0  34  0  0  32  5  10  12  40  33  12  36  39  44 1124
[1]  0  34  0  0  32  5  11  13  40  33  13  36  39  44 1116
[1]  0  34  0  0  32  5  12  14  40  33  14  36  39  44 1107
[1]  0  34  0  0  32  5  13  15  40  33  15  36  39  44 1099
[1]  0  34  0  0  32  5  14  16  40  33  16  36  39  44 1091
[1]  0  34  0  0  32  5  15  17  40  33  17  36  39  44 1083
[1]  0  34  0  0  32  5  16  18  40  33  18  36  39  44 1076
[1]  0  34  0  0  32  5  17  19  40  33  19  36  39  44 1068
[1]  0  34  0  0  32  5  18  20  40  33  20  36  39  44 1060
[1]  0  34  0  0  32  5  19  21  40  33  21  36  39  44 1053
[1]  0  34  0  0  32  5  20  22  40  33  22  36  39  44 1045
[1]  0  34  0  0  32  5  21  23  40  33  23  36  39  44 1038
```

```
[1]  0  34  0  0  32  5  22  24  40  33  24  36  39  44 1030
[1]  0  34  0  0  32  5  23  25  40  33  25  36  39  44 1023
[1]  0  34  0  0  32  5  24  26  40  33  26  36  39  44 1016
[1]  0  34  0  0  32  5  25  27  40  33  27  36  39  44 1009
[1]  0  34  0  0  32  5  26  28  40  33  28  36  39  44 1001
[1]  0  34  0  0  32  5  27  29  40  33  29  36  39  44 994
[1]  0  34  0  0  32  5  28  30  40  33  30  36  39  44 987
[1]  0  34  0  0  32  5  29  31  40  33  31  36  39  44 981
[1]  0  34  0  0  32  5  30  32  40  33  32  36  39  44 974
[1]  0  34  0  0  32  5  31  33  40  33  33  36  39  44 967
[1]  0  34  0  0  32  5  32  34  40  33  34  36  39  44 960
[1]  0  34  0  0  32  5  5  7  40  33  7  36  39  44 1166
[1]  0  34  0  0  32  5  5  8  40  33  7  36  39  44 1163
[1]  0  34  0  0  32  5  5  9  40  33  7  36  39  44 1160
[1]  0  34  0  0  32  5  5  10  40  33  7  36  39  44 1156
[1]  0  34  0  0  32  5  5  11  40  33  7  36  39  44 1153
[1]  0  34  0  0  32  5  5  12  40  33  7  36  39  44 1150
[1]  0  34  0  0  32  5  5  13  40  33  7  36  39  44 1147
[1]  0  34  0  0  32  5  5  14  40  33  7  36  39  44 1144
[1]  0  34  0  0  32  5  5  15  40  33  7  36  39  44 1141
[1]  0  34  0  0  32  5  5  16  40  33  7  36  39  44 1138
[1]  0  34  0  0  32  5  5  17  40  33  7  36  39  44 1134
[1]  0  34  0  0  32  5  5  18  40  33  7  36  39  44 1131
[1]  0  34  0  0  32  5  5  19  40  33  7  36  39  44 1128
[1]  0  34  0  0  32  5  5  20  40  33  7  36  39  44 1126
[1]  0  34  0  0  32  5  5  21  40  33  7  36  39  44 1123
[1]  0  34  0  0  32  5  5  22  40  33  7  36  39  44 1120
[1]  0  34  0  0  32  5  5  23  40  33  7  36  39  44 1117
[1]  0  34  0  0  32  5  5  24  40  33  7  36  39  44 1114
[1]  0  34  0  0  32  5  5  25  40  33  7  36  39  44 1111
[1]  0  34  0  0  32  5  5  26  40  33  7  36  39  44 1108
[1]  0  34  0  0  32  5  5  27  40  33  7  36  39  44 1106
[1]  0  34  0  0  32  5  5  28  40  33  7  36  39  44 1103
[1]  0  34  0  0  32  5  5  29  40  33  7  36  39  44 1100
[1]  0  34  0  0  32  5  5  30  40  33  7  36  39  44 1098
[1]  0  34  0  0  32  5  5  31  40  33  7  36  39  44 1095
[1]  0  34  0  0  32  5  5  32  40  33  7  36  39  44 1092
[1]  0  34  0  0  32  5  5  33  40  33  7  36  39  44 1090
[1]  0  34  0  0  32  5  5  34  40  33  7  36  39  44 1087
[1]  0  34  0  0  32  5  5  35  40  33  7  36  39  44 1085
[1]  0  34  0  0  32  5  5  36  40  33  7  36  39  44 1082
[1]  0  34  0  0  32  5  5  37  40  33  7  36  39  44 1080
[1]  0  34  0  0  32  5  5  38  40  33  7  36  39  44 1077
[1]  0  34  0  0  32  5  5  7  40  33  7  36  39  44 1166
[1]  0  34  0  0  32  5  5  7  40  33  7  36  39  44 1166
[1]  0  34  0  0  32  5  5  7  40  33  7  36  39  44 1166
[1]  0  34  0  0  32  5  5  7  40  33  8  36  39  44 1163
[1]  0  34  0  0  32  5  5  7  40  33  9  36  39  44 1159
```

```
[1]  0  34  0  0  32  5  5  7  40  33  10  36  39  44 1156
[1]  0  34  0  0  32  5  5  7  40  33  11  36  39  44 1152
[1]  0  34  0  0  32  5  5  7  40  33  12  36  39  44 1149
[1]  0  34  0  0  32  5  5  7  40  33  13  36  39  44 1146
[1]  0  34  0  0  32  5  5  7  40  33  14  36  39  44 1142
[1]  0  34  0  0  32  5  5  7  40  33  15  36  39  44 1139
[1]  0  34  0  0  32  5  5  7  40  33  16  36  39  44 1136
[1]  0  34  0  0  32  5  5  7  40  33  17  36  39  44 1133
[1]  0  34  0  0  32  5  5  7  40  33  18  36  39  44 1129
[1]  0  34  0  0  32  5  5  7  40  33  19  36  39  44 1126
[1]  0  34  0  0  32  5  5  7  40  33  20  36  39  44 1123
[1]  0  34  0  0  32  5  5  7  40  33  21  36  39  44 1120
[1]  0  34  0  0  32  5  5  7  40  33  22  36  39  44 1117
[1]  0  34  0  0  32  5  5  7  40  33  23  36  39  44 1114
[1]  0  34  0  0  32  5  5  7  40  33  24  36  39  44 1111
[1]  0  34  0  0  32  5  5  7  40  33  25  36  39  44 1108
[1]  0  34  0  0  32  5  5  7  40  33  26  36  39  44 1105
[1]  0  34  0  0  32  5  5  7  40  33  27  36  39  44 1102
[1]  0  34  0  0  32  5  5  7  40  33  28  36  39  44 1099
[1]  0  34  0  0  32  5  5  7  40  33  29  36  39  44 1096
[1]  0  34  0  0  32  5  5  7  40  33  30  36  39  44 1093
[1]  0  34  0  0  32  5  5  7  40  33  31  36  39  44 1091
[1]  0  34  0  0  32  5  5  7  40  33  32  36  39  44 1088
[1]  0  34  0  0  32  5  5  7  40  33  33  36  39  44 1085
[1]  0  34  0  0  32  5  5  7  40  33  34  36  39  44 1082
[1]  0  34  0  0  32  5  5  7  40  33   7  36  39  44 1166
[1]  0  34  0  0  32  5  5  7  40  33   7  36  39  44 1166
[1]  0  34  0  0  32  5  5  7  40  33   7  36  39  44 1166
```

1. 0 2. 34 3. 0 4. 0 5. 32 6. 5 7. 5 8. 7 9. 40 10. 33 11. 7 12. 36 13. 39 14. 44
1166.33559262078