

lista_5

July 16, 2018

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

```
In [77]: set.seed(1)
         library(triangle)
         library(igraph)
         library(hexbin)
         library(ggplot2)
```

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

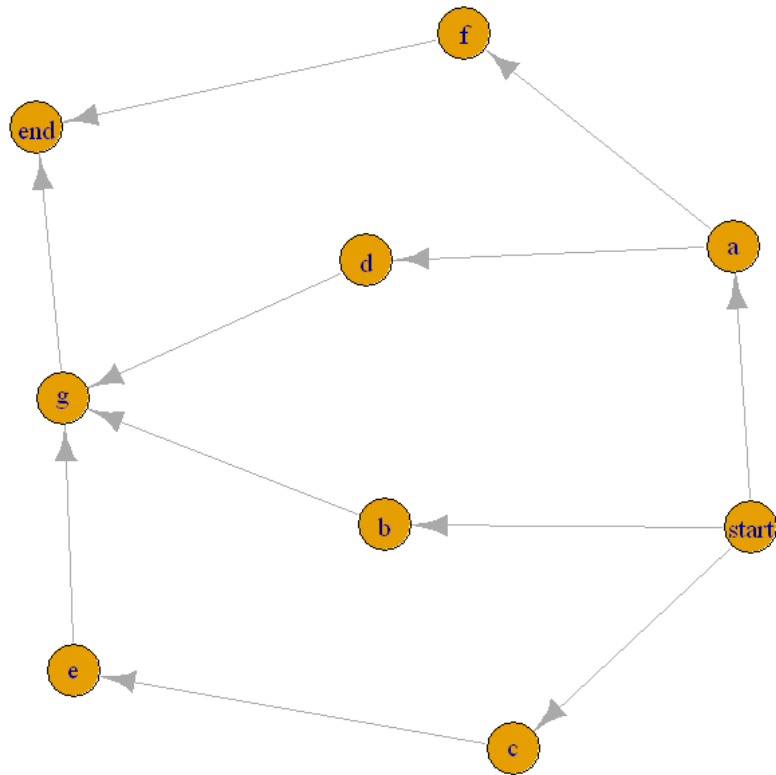
ativ	Min	Mprov	Max
A	1	2	4
B	5	6	7
C	2	4	5
D	1	3	4
E	4	5	7
F	3	4	5
G	1	2	3

Tabela 1 - Atividades

(start - A B C) (A - F D) (B - G) (C - E) (D - G) (E - G) (F - end) (G - end)

Tabela 2 - Redes

```
In [85]: 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 [2]: 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){
             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

```

11

'-start-C-E-G'

Obtenha um estimativa das probabilidades das atividades pertencerem ao caminho crítico

```

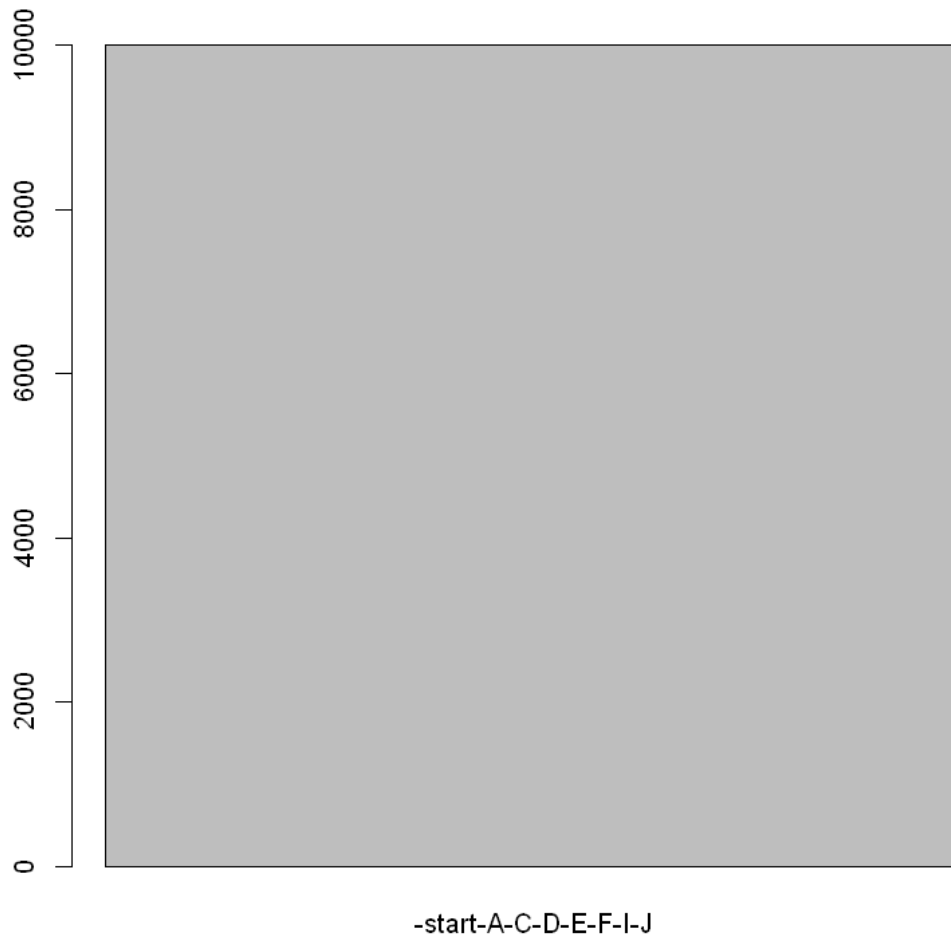
In [65]: resposta_final <- c()
caminho <- c()
for( h in 1:10000){
  precedencia <- list(c("start","A",rtriangle(1,1,4,2)),
                     c("start","B",rtriangle(1,5,7,6)),
                     c("start","C",rtriangle(1,2,5,4)),
                     c("A","F",rtriangle(1,3,5,4)),
                     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))

```



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

```

0.999681

Compare o resultado com aquele obtido pela aproximação PERT.

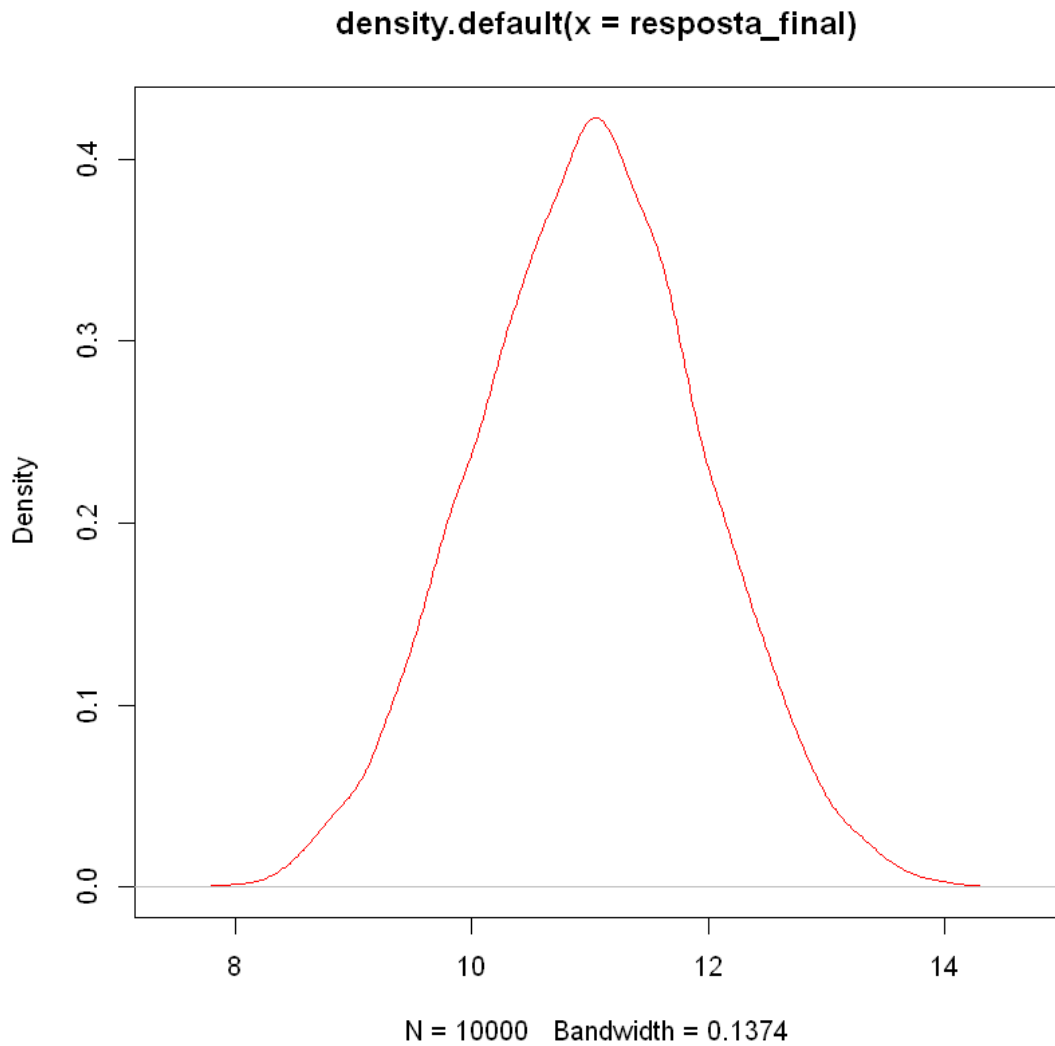
```
In [15]: resposta_final <- c()
  for( h in 1:10000){
    precedencia <- list(c("start","A",rtriangle(1,1,4,2)),
                        c("start","B",rtriangle(1,5,7,6)),
                        c("start","C",rtriangle(1,2,5,4)),
                        c("A","F",rtriangle(1,3,5,4)),
                        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)
```

11.0002021043627

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

```
In [16]: plot(density(resposta_final), col="red")
```



Gere todos agendamentos possíveis para o mínimo prazo.

start-B-G-end e start-C-E-G-end

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

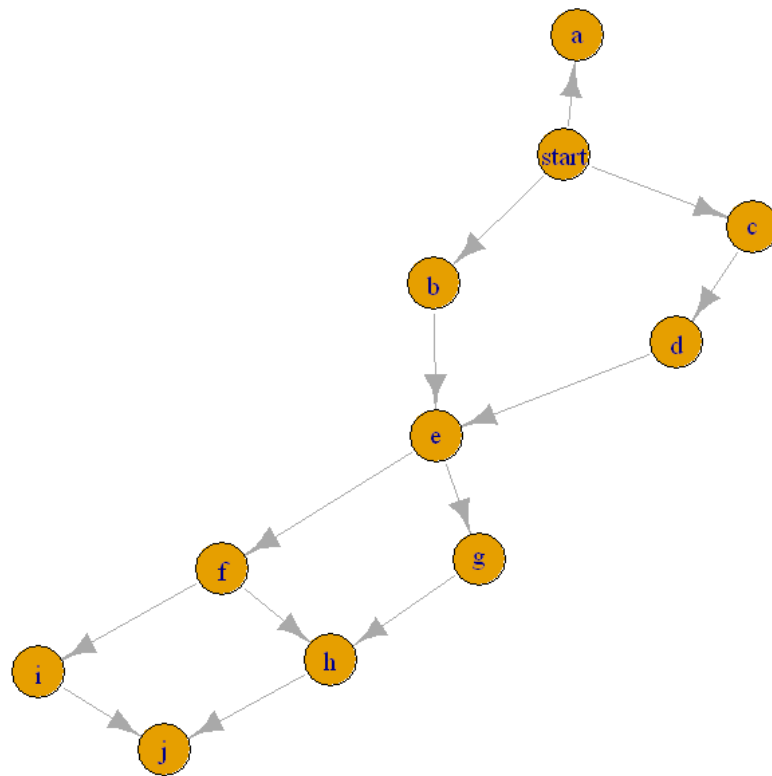
Atividade	Descrição	Pred	DMin	DMp	DMax	CMin	CMp	CMax
A	Obter materiais	START	2	4	18	300	450	600
B	Obter mão de obra	START	5	9	19	480	600	720
C	Escavar	START	4	10	28	3750	4500	5250
D	Colocar fundação	C	8	13	36	8400	9600	10800
E	Construir Estrutura	B,D	44	60	100	300000	312000	322500
F	Instalação Hidráulica	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

Atividade	Descrição	Pred	DMin	DMp	DMax	CMin	CMp	CMax
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 [60]: 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)
```

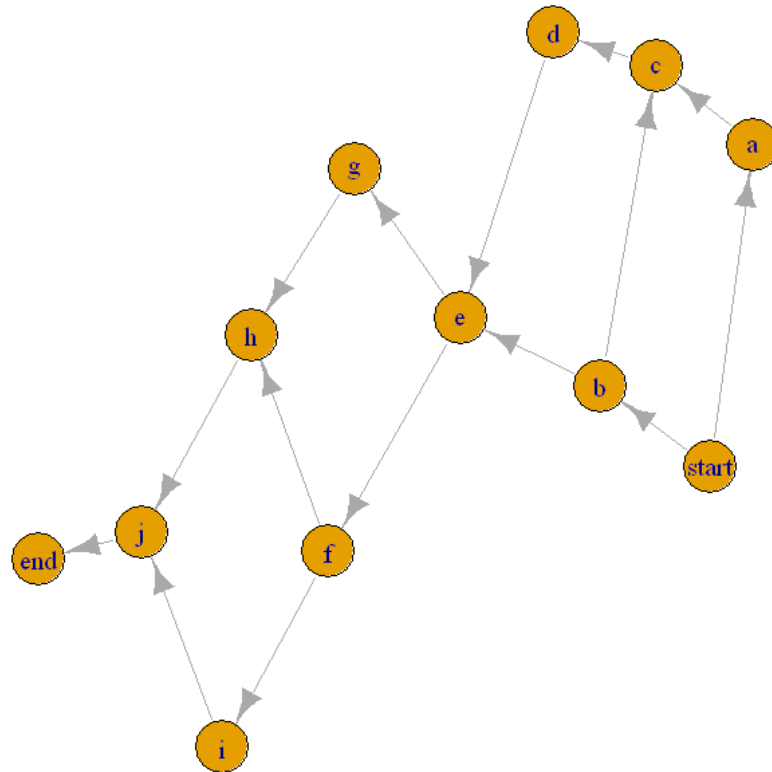


```
In [59]: g2 <- graph( edges=c('start','a', 'start','b', 'a', 'c','b','c','c','d',
                              'b','e','d','e','e','f','e','g','f','h','g','h','f',
```

```

        'i','h','j','i','j','j','end') )
plot(g2)

```



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

```

In [81]: 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)+

```

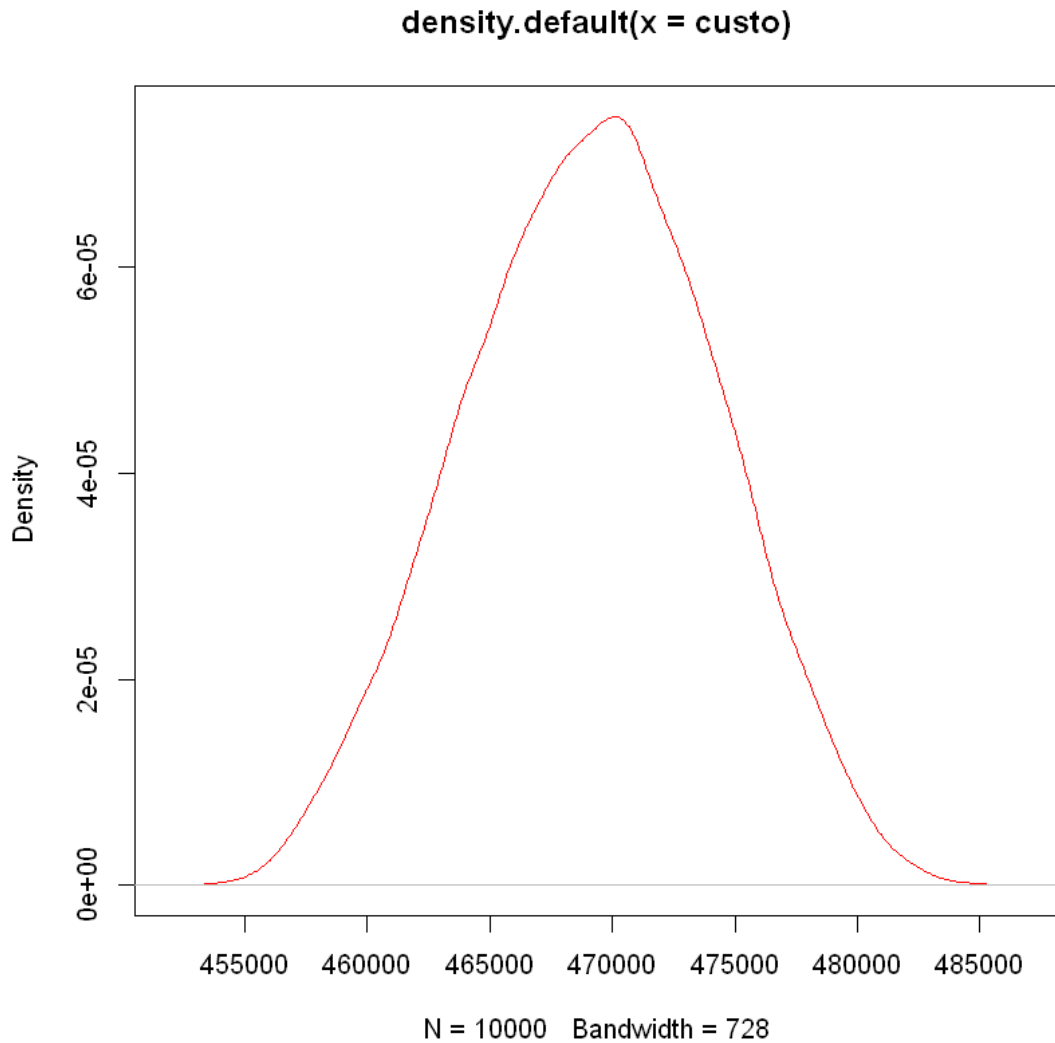


```

rtriangle(10000,360,540,450))
mean(custo)
plot(density(custo), col="red")

```

469137.333291898



```

In [19]: 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)),

```

```

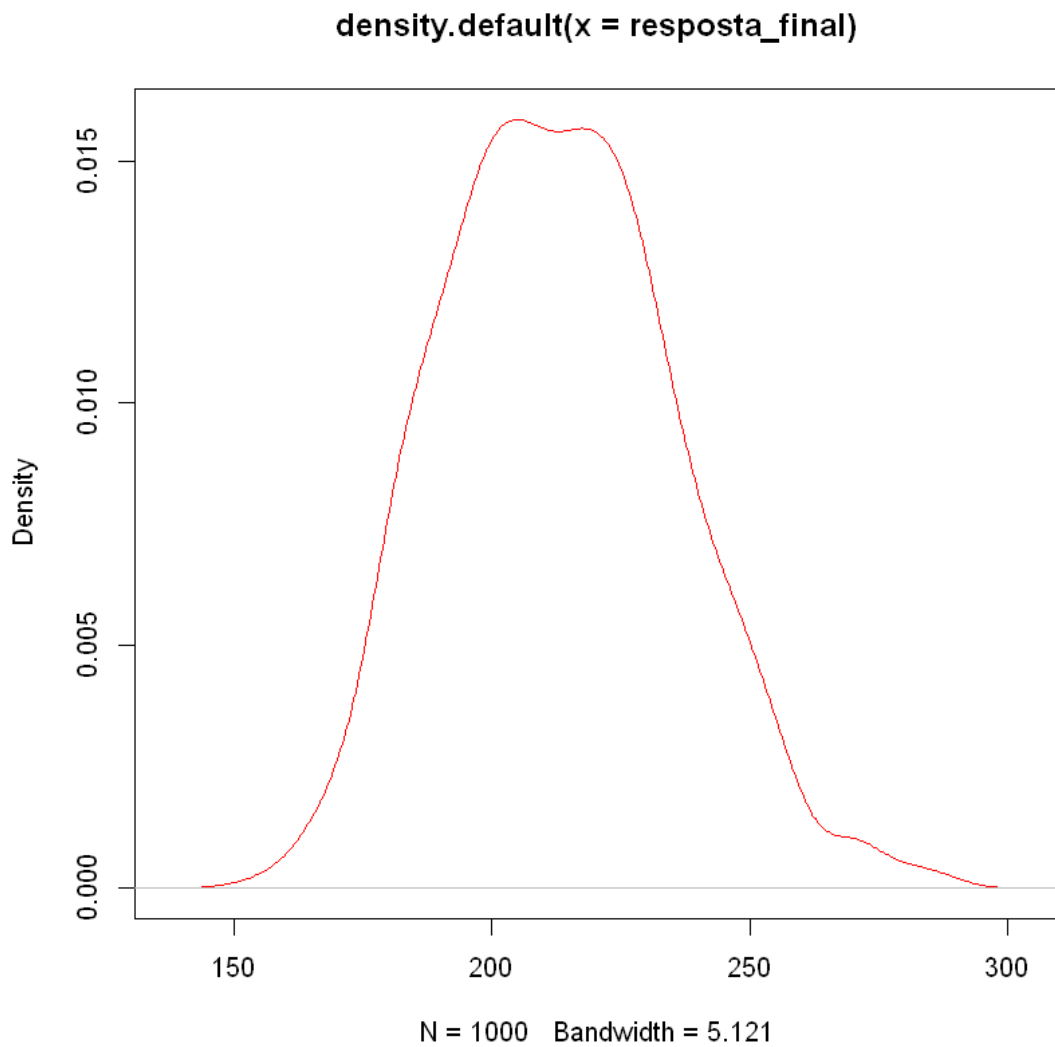
c("D","E",rtriangle(1,44,100,60)),
c("E","F",rtriangle(1,30,74,40)),
c("E","G",rtriangle(1,9,43,20)),
c("F","H",rtriangle(1,24,48,30)),
c("G","H",rtriangle(1,24,48,30)),
c("F","I",rtriangle(1,28,96,29)),
c("H","J",rtriangle(1,10,12,10)),
c("I","J",rtriangle(1,10,12,10)),
c("J","end",0),
c("J","end",0))

simula(precedencia)
}
mean(resposta_final)

```

213.036607108275

In [20]: plot(density(resposta_final), col="red")



```

In [69]: 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)

```

225.193428261673

Obtenha uma estimativa das probabilidades das atividades pertencerem ao caminho crítico

```

In [82]: 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)
}

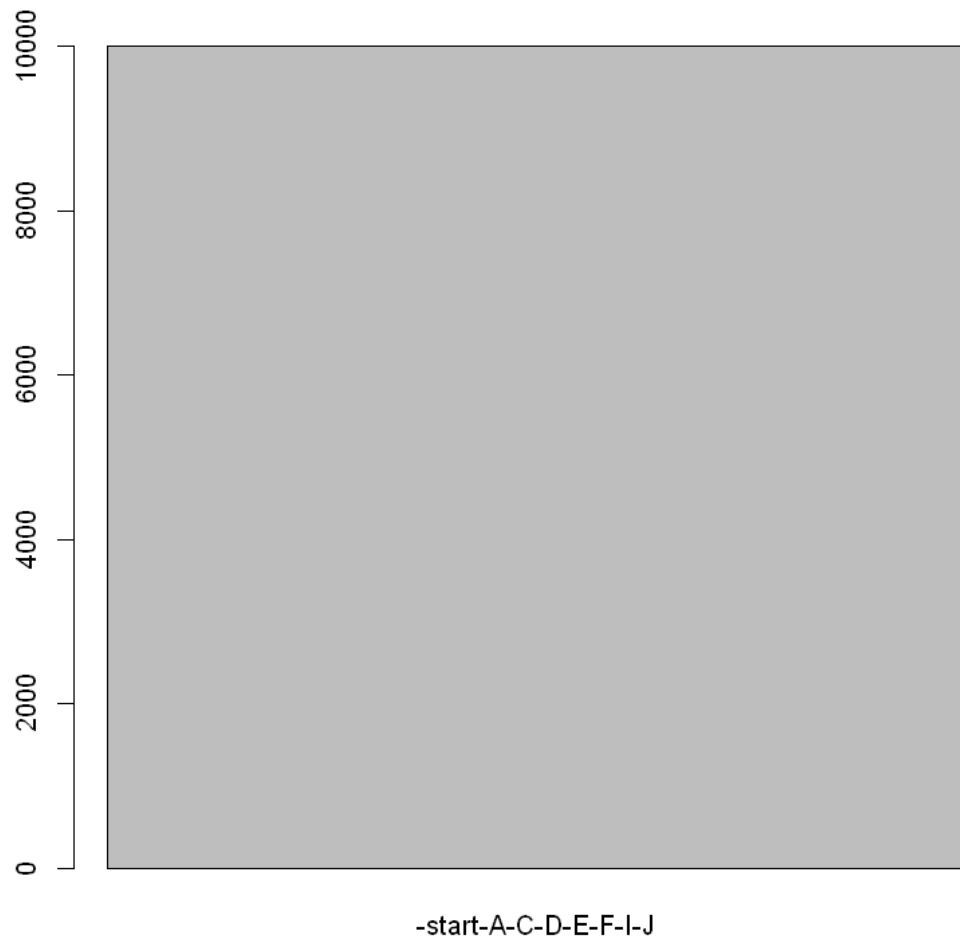
```

```

caminho <- c(caminho,ant)
}
mean(resposta_final)
barplot(table(caminho))

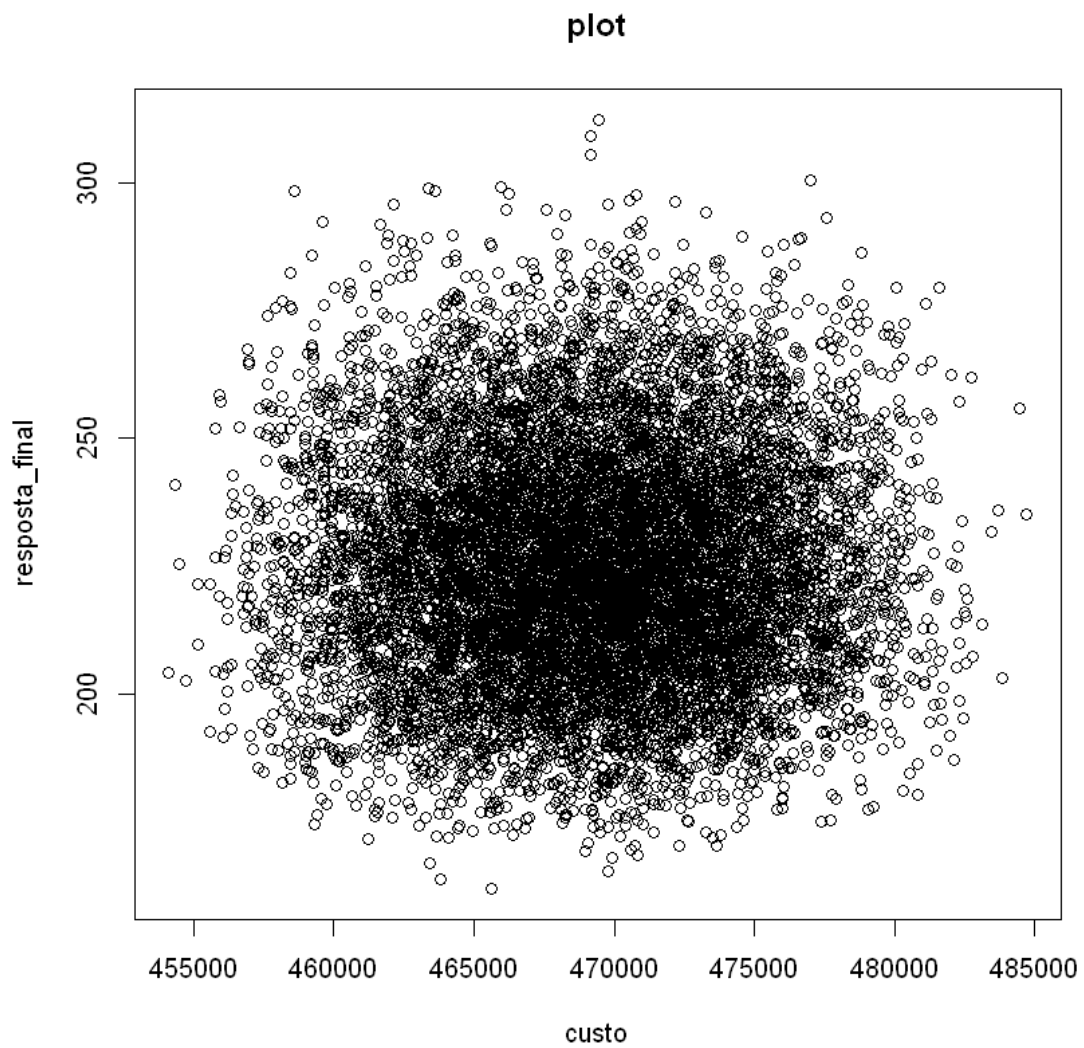
```

224.97739494658

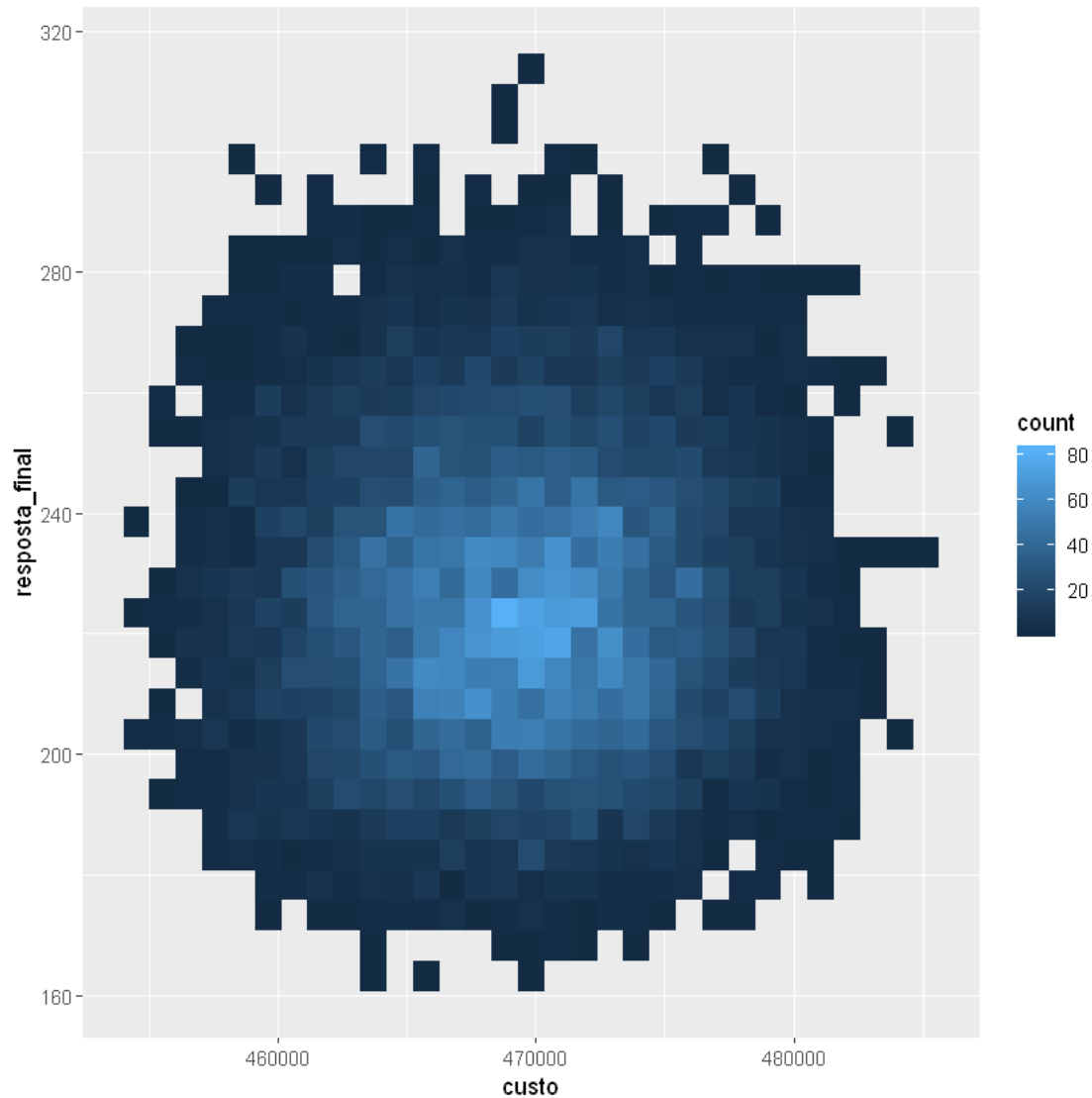


Faça um grafico de dispersão mostrando a correlação entre prazo e custo da obra.

```
In [83]: plot(custo,resposta_final, main="plot")
```



```
In [84]: df <- data.frame(custo,resposta_final)
         qqplot(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 círculo 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

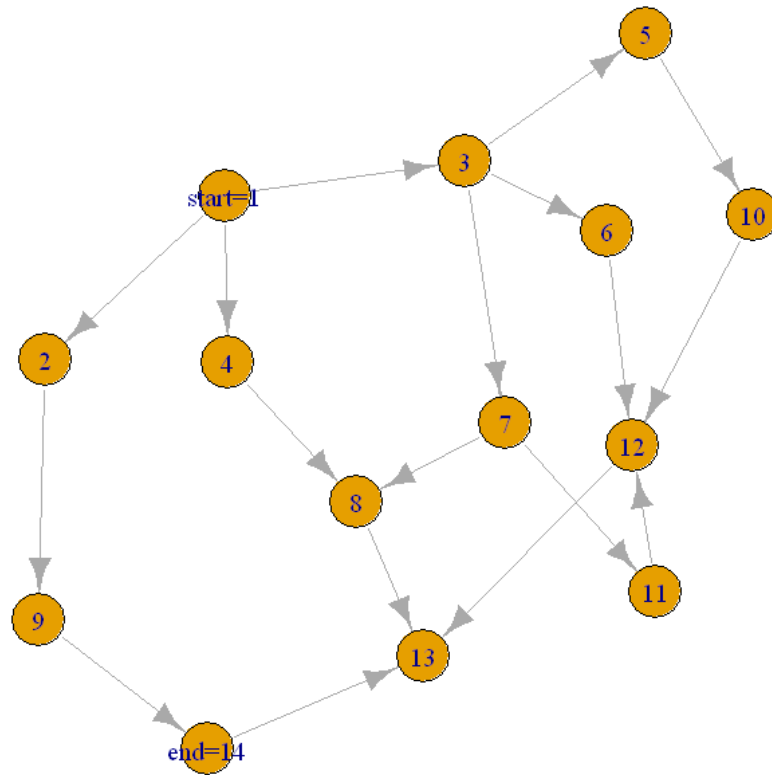
implementar uma função que verifica se o cronograma é válido

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 [228]: 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 [229]: 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 [230]: 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 [231]: npv <- function(inicio, valor, taxa){
    npv0 <- 0
    n <- length(inicio)

```

```

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

```

```

In [232]: 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 [233]: inicio <- inicio_cedo
  for(i in 1:n){
    novo <- inicio
    novo[[i]] = inicio_tarde[[i]]
    novo <- adia(novo,duracao,precedencia)
    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 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
1204.70605015516