# Escala

```
escala<-function(mes=mes,ano=ano,funcionarios=funcionarios)
{
if(mes<1|mes>12)#se o mês escolhido no argumento não for um valor que esteja
entre 1 e 12, a funcao nao executara
    {
    stop('mes deve ser um número entre 1 e 12')#erro para a condicao acima
    }
if(ano<1000) #se a pessoa colocar ano 18 ao inves de ano 2018
    {
    stop('ano deve conter 4 algarismos. Ex:2018')#a funcao não executara e
aparecera a messagem solicitando 4 algarismos
    }
library(lubridate)#pacote para carregar datas
#dependendo do mes escolhido, os número de dias será diferente. Os próximos
comandos são referentes a esse tópico
if(mes==1|mes==3|mes==5|mes==7|mes==8|mes==10|mes==12)#se o mês for
janeiro,março,maio,julho,agosto,outubro ou dezembro, a escala terá 31 dias
{
    inicio.do.mes<-ymd("2000-01-01")#comando para montar o primeiro dia do
mês com base no pacote lubridate no objeto inicio.do.mes
    final.do.mes<-ymd("2000-01-31")#comando para montar o último dia do mês
com base no pacote lubridate no objeto final.do.mes, nesse caso, o ultimo
dia do mes seria dia 31
    month(inicio.do.mes)<-mes#nomeia o mes do objeto inicio.do.mes de "mes"
que será determinado no argumento da funcao
    month(final.do.mes)<-mes#nomeia o mes do objeto final.do.mes de "mes"
que será determinado no argumento da funcao
    year(inicio.do.mes)<-ano#nomeia o ano do objeto inicio.do.mes de "ano"
que sera determinado no argumento da funcao
    year(final.do.mes)<-ano#nomeia o ano do objeto final.do.mes de "ano" que
sera determinado no argumento da funcao

    mes.selecionado<-
seq(from=as.Date(inicio.do.mes),to=as.Date(final.do.mes),by="day")#cria o
objeto mes.selecionado com a sequencia iniciando pelo objeto inicio.do.mes
(que indicara o dia 01/mes/ano) e finalizando pelo objeto final.do.mes (que
indicara o dia 31/mes/ano)
} else
if(mes==4|mes==6|mes==9|mes==11)#se o mês for abril, junho, setembro ou
novembro, a escala terá 30 dias
{
    inicio.do.mes<-ymd("2000-01-01")#comando para montar o primeiro dia do
mes com base no pacote lubridate no objeto inicio.do.mes
    final.do.mes<-ymd("2000-01-30")#comando para montar o ultimo dia do mes
com base no pacote lubridate no objeto final.do.mes, nesse caso, o ultimo
dia do mes sera dia 30
    month(inicio.do.mes)<-mes#nomeia o mes do objeto inicio.do.mes de "mes"
que será determinado no argumento da funcao
```

```
    month(final.do.mes)<-mes#nomeia o mes do objeto final.do.mes de "mes"
que será determinado no argumento da funcao
    year(inicio.do.mes)<-ano#nomeia o ano do objeto inicio.do.mes de "ano"
que sera determinado no argumento da funcao
    year(final.do.mes)<-ano#nomeia o ano do objeto final.do.mes de "ano" que
sera determinado no argumento da funcao

    mes.selecionado<-
seq(from=as.Date(inicio.do.mes),to=as.Date(final.do.mes),by="day")#cria o
objeto mes.selecionado com a sequencia iniciando pelo objeto inicio.do.mes
(que indicara o dia 01/mes/ano) e finalizando pelo objeto final.do.mes (que
indicara o dia 30/mes/ano)
} else
#a situacao abaixo serve para o mes de fevereiro, que pode ter 28 ou 29
dias. Se o ano for multiplo de 4, isso indica que o ano é bissexto, entao o
mes tera 29 dias, se nao for multiplo de 4, entao o mes tera 28 dias
if(mes==2&&ano%%4==0)#rodar a situacao abaixo se o mes for fevereiro e o ano
for multiplo de 4, logo o mes tera 29 dias
{
    inicio.do.mes<-ymd("2000-02-01")#comando para montar o primeiro dia do
mes com base no pacote lubridate no objeto inicio.do.mes
    final.do.mes<-ymd("2000-02-29")#comando para montar o ultimo dia do mes
com base no pacote lubridate no objeto final.do.mes, nesse caso, o ultimo
dia do mes sera dia 29
    year(inicio.do.mes)<-ano#nomeia o ano do objeto inicio.do.mes de "ano"
que sera determinado no argumento da funcao
    year(final.do.mes)<-ano#nomeia o ano do objeto final.do.mes de "ano" que
sera determinado no argumento da funcao
    mes.selecionado<-
seq(from=as.Date(inicio.do.mes),to=as.Date(final.do.mes),by="day")#cria o
objeto mes.selecionado com a sequencia iniciando pelo objeto inicio.do.mes
(que indicara o dia 01/mes/ano) e finalizando pelo objeto final.do.mes (que
indicara 29/mes/ano)
}
if(mes==2&&ano%%4!=0)#se o mes for fevereiro e o ano nao for multiplo de 4,
o mes tera 28 dias
{
    inicio.do.mes<-ymd("2000-02-01")#comando para montar o primeiro dia do
mes com base no pacote lubridate no objeto inicio.do.mes
    final.do.mes<-ymd("2000-02-28")#comando para montar o ultimo dia do mes
com base no pacote lubridate no objeto final.do.mes, nesse caso, o ultimo
dia do mes sera dia 28
    year(inicio.do.mes)<-ano#nomeia o ano do objeto inicio.do.mes de "ano"
que sera determinado no argumento da funcao
    year(final.do.mes)<-ano#nomeia o ano do objeto final.do.mes de "ano" que
sera determinado no argumento da funcao

    mes.selecionado<-
seq(from=as.Date(inicio.do.mes),to=as.Date(final.do.mes),by="day")#cria o
objeto mes.selecionado com a sequencia iniciando pelo objeto inicio.do.mes
```

```
(que indicara o dia 01/mes/ano) e finalizando pelo objeto final.do.mes (que
indicara 28/mes/ano)
}
#os objetos abaixo servirao para preencher o data frame da escala
dias.do.mes = day(mes.selecionado) #vetor separando os dias do objeto
mes.selecionado
semana=wday(mes.selecionado)#vetor separando os dias da semana do objeto
mes.selecionado
dia<-NA#vetor dia contendo NA
noite<-NA#vetor noite contendo NA
noite.2<-NA#vetor noite.2 contendo NA

escala<-data.frame(dias.do.mes,dia,noite,noite.2) #data frame com os objetos
criados anteriormente (semana sera adicionado ao objeto no final da funcao)
#o data frame esta pronto para ser preenchido

#a partir deste pronto comeca o preenchimento, que sera de uma forma para 5
funcionarios e de outra para mais que 5, uma vez que para 5 funcionarios, os
dias que eles podem trabalhar e respeitam o descanso sao especificos e o
preenchimentos deve ser feito em sequencia de linhas para respeitar o
descanso, ja para mais funcionarios, a sequencia utilizada foi em colunas
para que o r escolha o dia que a pessoa trabalhara
if(funcionarios==5)#se o numero de funcionarios for igual a 5
{
    if(length(dias.do.mes)==28|length(dias.do.mes)==29)#para 28 e 29
funcionarios
    {
    sequencia = rep (1:funcionarios,times=6)#e criado o objeto sequencia com
repeticao de 1 a 5 funcionarios 6 vezes
    dia.15<-rep(c(1,3,5,2,4),times=3)#ate o dia 15 da coluna dia sao
adicionados os funcionarios 1,3,5,2 e 4 (3 vezes). Dessa forma, sera seguido
o criterio descanso e o criterio distribuicao de diurno e noturno
    noite.15<-rep(c(2,4,1,3,5),times=3)#este objeto faz o mesmo do objeto
acima, mas na coluna noite na sequencia 2,4,1,3,5
#os dois comando acima representam na pratica a sequencia de 1 a 5
preenchendo a tabela por linha, então na primeira linha tera 1 na coluna dia
e 2 na coluna noite, na segunda linha, 3 na coluna dia e 4 na coluna noite,
na terceira linha, 5 na coluna dia e q na coluna noite, e assim por diante
ate o dia 15
    escala$dia[c(1:15)]<-dia.15#neste comando o objeto dia.15 e adicionado
na coluna dia da escala de 1 a 15 dias
    escala$noite[c(1:15)]<-noite.15#o mesmo que acima, mas na coluna noite
#a partir dessa etapa a coluna noite.2 passa a ser preenchida tambem
    escala$dia[c(16:25)]<-
sequencia[c(seq(from=1,to=7,by=2),seq(from=10,to=14,by=2),seq(from=17,to=21,
by=2))]#da linha 16 a 25, serao retirados do objeto sequencia que tem a
repeticao de funcionarios em uma ordem pre determinada de acordo com o
descanso dos funcionarios
    escala$noite[c(16:25)]<-
sequencia[c(seq(from=2,to=8,by=2),seq(from=11,to=15,by=2),seq(from=18,to=22,
by=2))]# o mesmo que acima para a coluna noite
```

```
        if(length(dias.do.mes)==28)#se o mes for de 28 dias, faltarao mais
plantoes para os funcionarios do que para 29 dias, entao mais linhas da
coluna noite.2 deverao ser preenchidas
        {
        noite.2<-sequencia[c(9,16,23,30)]#aqui cria-se o objeto noite.2 com
as pessoas pre determinadas que preencherao os dias da coluna noite.2
        escala$noite.2[c(19,22,25,28)]<-noite.2#o objeto criado acima
preenchera datas tambem pre determinadas

        escala$dia[c(26:28)]<-sequencia[seq(from=24,to=28,by=2)]#nesta etapa
preenche-se tambem dias especificos para 28 dias
        escala$noite[c(26:28)]<-sequencia[seq(from=25,to=29,by=2)]#o mesmo
que acima
        }
        if(length(dias.do.mes)==29)#para 29 dias, sera necessario o uso de
menos dias da coluna noite.2
        {
        noite.2<-sequencia[c(9,16)]#selecao das pessoas para a coluna
noite.2
        escala$noite.2[c(19,22)]<-noite.2#preenchendo a coluna noite.2 nas
posicoes pre determinadas

        escala$dia[c(26:29)]<-sequencia[seq(from=23,to=29,by=2)]#preenchendo
os ultimos dias com pessoas especificos na coluna dia
        escala$noite[c(26:29)]<-sequencia[seq(from=24,to=30,by=2)]#o mesmo
que acima para a coluna noite
        }

    }
    if(length(dias.do.mes)==31|length(dias.do.mes)==30)#se o numero de dias
for 30 ou 31 dias, o preenchimento é outro
    {
    func = seq (1:funcionarios)#primeiro cria-se um objeto com a sequencia
dos funcionarios, sem repeticao, chamado func
    rep.func = rep (func, len = nrow (escala))#nesse objeto esta a repeticao
de func de acordo com o numero de dias do mes

    escala$dia = rep.func#preencher a escala dia com o rep.func
#a partir dessa etapa o objetivo e puxar a posicao de cada funcionario na
coluna dia ja preenchida na etapa anterior
    col = round ((nrow (escala)/funcionarios)) + 1#objeto col calcula o
numero de dias dividido pelo numero de funcionarios mais 1. esse objeto sera
usado para determinar o numero de colunas da matriz que mostrara os dias que
as pessoas ja estao posicionadas na coluna dia
    posicao = matrix (NA, funcionarios, col)#matriz que tera as posicoes dos
funcionarios. cada linha representara um funcionario
#comeca entao o preenchimento da matriz "posicao" para cada funcionario
        for (i in 1:funcionarios)#ciclo realizado para o funcionario 1 ate o
ultimo
        {
```

```
        teste = which (escala$dia == i)#o objeto teste procura em quais
posicoes o funcionario i esta na coluna dia da escala
        if (length (teste) != col)#essa etapa foi criada para garantir que o
numero de colunas seja igual ao numero de dias que serao encontrados no
objeto teste
            {
            teste = c (teste, rep(NA,length(posicao[i,])-length(teste)))#se
o numero de colunas for diferente do numero de dias do objeto teste,
adiciona-se NA ao objeto teste
            }
        posicao[i, ] =  teste#na matriz posicao acrescentam-se os dias
encontrados no objeto teste
        }
#a partir desse momento sera feito o calculo dos dias que o funcionario pode
trabalhar com base nos dias que ele caiu na coluna dia
    col2 = (2* length (func)) + 2#numero de colunas que garantira espaco
necessario para preencher a matriz
    descanso = matrix (NA, length (func), col2+2)#matriz "descanso" com os
dias que a pessoa pode trabalhar respeitando o descanso dela, contendo NA e
com o numero de linhas igual ao numero de funcionarios. Cada linha
representa um funcionario

        k=1#contador que representa a coluna da matriz descanso

        for (i in 1:nrow (posicao))#para cada linha i da matriz posicao
        {
              for (j in 1:ncol (posicao))#para cada coluna j da matriz
posicao
            {
            descanso [i, k] = posicao[i, j] - 3#acrescentar na matriz
descanso na posicao da linha i, coluna k, o dia que estava na linha i,
coluna j subtraido de 3, ou seja, a pessoa que estava no dia 10 diurno,
podera trabalhar no dia 7 noturno
            k = k + 1#pular para a proxima coluna da matriz descanso
            descanso [i, k] = posicao[i, j] + 2#acrescentar na linha i
coluna k+1, o dia que estiver na posicao i coluna j adicionado de 2, ou
seja, a pessoa trabalhou no dia 10 diurno, pode voltar no dia 12 noturno
            k = k + 1 #pular para a proxima coluna da matriz descanso
            }
            k = 1#zera-se o contador k para continuar o ciclo
        }
#finaliza-se o preenchimento da matriz descanso
    descanso[descanso<"1"]<-NA#a matriz descanso terá numeros negativos e
numeros acima do numero de dias do mes. Nessa linha excluem-se os numeros
negativos
    descanso[descanso>nrow(escala)]<-NA#excluem-se os valores acima do
numero de dias do mes
    frequencia<-table(descanso)#objeto frequencia conta quantas ocorrencias
tem de cada dias na matriz descanso
    frequencia.2<-names(frequencia)#frequencia .2 isola os dias do objeto
frequencia
```

```
    frequencia.2<-as.numeric(frequencia.2)#o objeto gerado acima é
classificado como character, sendo necessario converte-lo para numerico para
prosseguir

#a partir desse momento, comeca o preenchimento de fato da escala
        for (x in 1:funcionarios) #ciclo se inicia para o contador x que vai
do funcionario 1 ate o ultimo funcionario
            {
            dias<-frequencia.2 %in% descanso[x,]#o objeto "dias" é um teste
logico que verifica na linha x (x=funcionario), os dias que a pessoa esta na
matriz descanso
            dias.2<-frequencia.2[dias]#o objeto "dias.2" mostra que dias sao
TRUE no objeto "dias", ou seja, ele guarda no objeto dias.2 os dias que a
pessoa x esta na matriz descanso na linha x
            escala[dias.2, "noite"] = x#esses dias sao unicos para 5
funcionarios, entao obrigatoriamente quem estiver na planilha descanso sera
colocado no mesmo dia na coluna noite
            }
    if(length(dias.do.mes)==31)#para meses com 31 dias é necessario
preencher alguns dias da coluna noite.2. Para que o descanso seja
respeitado, foi necessario pre determinar os dias da coluna noite.2
        {
        escala$dia[1]<-3#na coluna dia posicao 1 deve ser encaixado o
funcionario 3
        escala$noite.2[1]<-1#o mesmo que acima mas o funcionario 1 na
posicao 1 da noite.2
        escala$noite[26]<-2# e assim por diante, sendo o numero a direita da
seta a pessoa que entrara no dia e o colchete sendo o dia que a pessoa
entrara na coluna noite, dia ou noite.2
        escala$dia[27]<-4#o mesmo que acima
        escala$noite[29]<-5#o mesmo que acima
        escala$dia[30]<-2#o mesmo que acima
        escala$noite.2[28]<-2#o mesmo que acima
        escala$noite.2[31]<-5#o mesmo que acima
#as etapas anteriores resolve o problema de decanso para 5 funcionarios, uma
vez que o numero de pessoas para distribuir ao longo do mes e pequeno, tendo
menos opcoes para preencher a escala
        }
    }
}
if(funcionarios!=5)#a proxima etapa serve para mais que 5 funcionarios
{
    func = seq (1:funcionarios)#primeiro cria-se um objeto com a sequencia
dos funcionarios, sem repeticao, chamado func
    rep.func = rep (func, len = nrow (escala))#nesse objeto esta a repeticao
de func de acordo com o numero de dias do mes

    escala$dia = rep.func#preencher a escala dia com o rep.func
#a partir dessa etapa o objetivo e puxar a posicao de cada funcionario na
coluna dia ja preenchida na etapa anterior
```

```
    col = round ((nrow (escala)/funcionarios)) + 1#objeto col calcula o
numero de dias dividido pelo numero de funcionarios mais 1. esse objeto sera
usado para determinar o numero de colunas da matriz que mostrara os dias que
as pessoas ja estao posicionadas na coluna dia
    posicao = matrix (NA, funcionarios, col)#matriz que tera as posicoes dos
funcionarios. cada linha representara um funcionario
#comeca entao o preenchimento da matriz "posicao" para cada funcionario
        for (i in 1:funcionarios)#ciclo realizado para o funcionario 1 ate o
ultimo
        {
        teste = which (escala$dia == i)#o objeto teste procura em quais
posicoes o funcionario i esta na coluna dia da escala
        if (length (teste) != col)#essa etapa foi criada para garantir que o
numero de colunas seja igual ao numero de dias que serao encontrados no
objeto teste
            {
            teste = c (teste, rep(NA,length(posicao[i,])-length(teste)))#se
o numero de colunas for diferente do numero de dias do objeto teste,
adiciona-se NA ao objeto teste
            }
        posicao[i, ] =  teste#na matriz posicao acrescentam-se os dias
encontrados no objeto teste
        }
#a partir desse momento sera feito o calculo dos dias que o funcionario pode
trabalhar com base nos dias que ele caiu na coluna dia
    col2 = (2* length (func)) + 2#numero de colunas que garantira espaco
necessario para preencher a matriz
    descanso = matrix (NA, length (func), col2+2)#matriz "descanso" com os
dias que a pessoa pode trabalhar respeitando o descanso dela, contendo NA e
com o numero de linhas igual ao numero de funcionarios. Cada linha
representa um funcionario

        k=1#contador que representa a coluna da matriz descanso

        for (i in 1:nrow (posicao))#para cada linha i da matriz posicao
        {
              for (j in 1:ncol (posicao))#para cada coluna j da matriz
posicao
            {
            descanso [i, k] = posicao[i, j] - 3#acrescentar na matriz
descanso na posicao da linha i, coluna k, o dia que estava na linha i,
coluna j subtraido de 3, ou seja, a pessoa que estava no dia 10 diurno,
podera trabalhar no dia 7 noturno
            k = k + 1#pular para a proxima coluna da matriz descanso
            descanso [i, k] = posicao[i, j] + 2#acrescentar na linha i
coluna k+1, o dia que estiver na posicao i coluna j adicionado de 2, ou
seja, a pessoa trabalhou no dia 10 diurno, pode voltar no dia 12 noturno
            k = k + 1 #pular para a proxima coluna da matriz descanso
            }
            k = 1#zera-se o contador k para continuar o ciclo
        }
```

```
#finaliza-se o preenchimento da matriz descanso
    descanso[descanso<"1"]<-NA#a matriz descanso terá numeros negativos e
numeros acima do numero de dias do mes. Nessa linha excluem-se os numeros
negativos
    descanso[descanso>nrow(escala)]<-NA#excluem-se os valores acima do
numero de dias do mes
    frequencia<-table(descanso)#objeto frequencia conta quantas ocorrencias
tem de cada dias na matriz descanso

    unico<-frequencia[frequencia=="1"]#objeto unico busca no objeto
frequencia que dia ocorre apenas uma vez no objeto frequencia, ou seja,
apenas uma pessoa pode trabalhar nesses dias
    duplicado<-frequencia[frequencia>"1"]#o objeto duplicado busca no objeto
frequencia que dia ocorre mais de uma vez no objeto frequencia, ou seja,
mais de uma pessoa pode trabalhar nesses dias
    unico.2<-names(unico)#isola os dias do objeto unico
    duplicado.2<-names(duplicado)#isola os dias do objeto duplicado
    unico.2<-as.numeric(unico.2)#como a classe do objeto gerado acima e
character, deve-se converter para numerico
    duplicado.2<-as.numeric(duplicado.2)#o mesmo que acima

    keep<-matrix(NA,nrow=col2,ncol=col2)#essa matriz foi gerada para guardar
valores dos dias que já foram preenchidos a cada ciclo de funcionarios x,
com um coluna de linhas e colunas suficientes para serem preenchidas

    for (x in 1:funcionarios)#ciclo para preencher a matriz "keep" para o
contador x de 1 até o ultimo funcionario. Cara coluna ira representar um
funcionario e o que aparesera nessa matriz e o dia que esse funcionarios foi
encaixado
    {
        dias.unico<-unico.2%in%descanso[x,]#objeto dias.unico que faz um
teste logico para verificar os dias que apenas a pessoa x podera trabalhar
na matriz descanso
        dias.unico.2<-unico.2[dias.unico]#objeto dias.unico.2 que mostra os
dias que apenas a pessoa x podera trabalhar
        dias<-duplicado.2 %in% descanso[x,]#objeto dias em que verifica
quais dias a pessoa pode trabalhar que mais de uma pessoa poderia trabalhar
tambem
        dias.2<-duplicado.2[dias]#objeto dias.2 mostra os dias que a pessoa
pode trabalhar, mas que mais uma pessoa tambem pode trabalhar
        escala[dias.unico.2, "noite"] = x#preenche a planilha com os dias
unicos, pois só ela pode trabalhar naquele dia
        descanso[x,][descanso[x,]==dias.unico.2]<-NA#insere NA no dia que
foi preenchido
        dias.3<-c(dias.2[1],sample(c(dias.2[2],dias.2[3]),1),
sample(c(dias.2[4],dias.2[5]),1),sample(c(dias.2[6],dias.2[7]),1),sample(c(d
ias.2[8],dias.2[9]),1),sample(c(dias.2[10],dias.2[11]),1))#objeto dias.3 com
os dias gerados no objeto dias.2, em que se sorteia um dos dias para cada
coluna, uma vez que se a pessoa esta em algum dia da coluna [,2] da planilha
descanso, ela não podera ficar no dia da coluna[,3] da mesma planilha para
```

```
possibilitar uma pausa maior entre um plantao e outro
        dias.3 <- dias.3[!is.na(dias.3)]#remocao de NA do objeto dias.3,
pois como a matriz descanso tambem possui NA, no sorteio, pode ocorrer de
pegar um NA

        guardado<-dias.3%in%keep#o objeto guardado busca na matriz keep se
os dias que foram sorteados no objeto dias.3(ou seja, os dias em que a
pessoa sera colocada) já ocorreram na matriz keep, ou seja, se ocorreram,
esses dias nao serao utilizados na coluna noite,para evitar sobreposicao de
pessoas, se não foi utilizado, então o vetor dias.3 podera ser utilizado na
coluna noite
        guardado.2<-dias.3[guardado==FALSE]#guardado.2 busca os valores que
deram FALSE no objeto guardado. Esses sao os dias de interesse para
preencher a escala na coluna noite, pois sao os dias que ainda nao foram
preenchidos
        guardado.3<-dias.3[guardado]#esse objeto guarda os dias que deram
TRUE no objeto guardado, ou seja, ja ocorreram na matriz keep, então nao
podem preencher a coluna noite, mas podem prencher a coluna noite.2
        conteudo<-length(guardado.3)#o conteudo guarda o comprimento de
guardado.3. Se o comprimento for igual 0, isso significa que todos os dias
escolhidos podem preencher a coluna noite, pois todos deram FALSE na busca
pelos dias na matriz keep, ou seja, eles ainda não tiveram nenhuma
ocorrencia na distribuicao. Porem se for diferente de 0, significa que tem
dias que a pessoa nao podera preencher a coluna noite, preenchendo entao a
coluna noite.2
        if(conteudo!=0)#dessa forma, se o comprimento de conteudo for
diferente de 0...
            {
            escala[guardado.2,"noite"]=x#a pessoa x sera inserida nos dias
do objeto guardado.2 na coluna noite, ou seja, sao os dias que ainda nao
foram preenchidos na coluna noite
            escala[guardado.3,"noite.2"]=x#e a mesma pessoa x sera inserida
na coluna noite.2 para os dias contidos no objeto guardado.3, pois ja esta
ocupado o mesmo dia na coluna noite, evitando que sobrescreva a pessoa que
foi adicionada anterioromente
            if(length(guardado.2)!=0)#se o comprimento de guardado.2 for
diferente de 0
            {
            keep[1:length(guardado.2),x]<-guardado.2 # os dias que foram
utilizados serão salvos na matriz keep
            }
        }
        if(conteudo==0) #se o conteudo for igual a 0
                {
                escala[dias.3, "noite"] = x# entao nao houve ocorrencia
ainda para nenhum dos dias selecionados, então a pessoa x pode ser inserida
nesses dias na coluna noite, sem sobrescrever ninguem
                    if(length(dias.3)!=0)#se o comrpimento de dias.3 for
diferente de 0
                    {
                    keep[1:length(dias.3),x]<-dias.3 # os dias que foram
```

```
utilizados serão salvos na matriz keep
                    }
                    for(n in 1:length(dias.3))#ciclo para tirar os valores
que ja preencheram a escala
                    {
                        dia<-dias.3[n]#dia busca os dias que foram
selecionados em dias.3 que irao para a coluna noite
                        procura.dia<-dia%in%descanso[x,]#procura.dia busca o
dia na matriz descanso
                        procura.dia.2<-dia[procura.dia]#isola o dia que tem
no descanso
                            if(length(procura.dia.2)!=0)#se o comprimento de
procura.dia.2 for diferente de 0, então o dia sera retirado da tabela
descanso
                            {
                            descanso[x,][descanso[x,]==dia]<-NA#insere NA no
dia que foi preenchido
                            descanso[x,][descanso[x,]==(dia-1)]<-NA#insere
NA no dia anterior, se houver,uma vez que não respeitara o descanso
                            descanso[x,][descanso[x,]==(dia+1)]<-NA#insere
NA no dia seguinte,se fouver, uma vez que não respeitara o descanso
                            }
                    }
                }
        }

#apos esse primeiro preenchimento, e feita a contagem de plantoes de cada um
    total.1<-table(unlist(subset(escala[,(2:4)])))#total.1 e o objeto que
ira guardar a contagem de ocorrencias de cada funcionarios nas colunas
dia,noite e noite.2
    numero.plantoes<-max(total.1)#esse objeto mostra qual e o numero mais
alto de total.1
    falta.preencher<-numero.plantoes-total.1#falta.preencher e o objeto que
calcula quantos plantoes faltam para cada funcionario para chegar ao numero
maximo de plantao calculado no objeto anterior
    teste.total<-cbind(total.1,falta.preencher)#teste.total une em uma
tabela o objeto total.1 e o objeto falta.preencher
    escala$noite[is.na(escala$noite)]<-0#para que o proximo passo funcione é
necessario que os NA's da coluna noite sejam transformados em 0
    escala$noite.2[is.na(escala$noite.2)]<-0#o mesmo que acima na coluna
noite.2

    for (z in 1:funcionarios)#ciclo para o contador z do funcionario 1 ate o
ultimo
    {
        while(teste.total[z,2]!=0)#enquanto a coluna falta.preencher do
funcionario z for diferente de 0, o proximo passo deve ocorrer para que
garante que todas as pessoas tenham o mesmo numero de plantoes
        {
        dias<-duplicado.2 %in% descanso[z,]#mesmo teste utilizado no
```

```
preenchimento anterior, mas dependente da tabela teste.total. Buscar os dias
duplicados da pessoa x na tabela descanso
        dias.2<-duplicado.2[dias]#mostra quais dias deram TRUE no teste
anterior
        dias.3<-c(sample(dias.2[2:length(dias.2)],1))#sorteio de qualquer um
dos dias que a pessoa pode trabalhar
        while(dias.3%in%keep[,z])#se esse dia estiver na tabela keep, ele
nao podera ser usado, pois ja tem uma pessoa naquele dia na coluna noite
        {
            dias.3<-c(sample(dias.2[2:length(dias.2)],1))#se for encontrado
que o dia sorteado ja esta na tabela keep, devera ocorrer um novo sorteio,
até que seja sorteado um dia que nao estiver na tabela keep
        }
#para que o ciclo funcione, é necessario transformar os NA's das colunas
noite e noite.2 em 0 para funcionar a logica do if que vem em sequencia e
posteriormente retornar para NA para fazer a contagem com a exclusao do 0
        escala$noite[is.na(escala$noite)]<-0#dessa forma, toda vez que um
funcionario rodas no ciclo, os NA's sao transformados em 0 na coluna noite
        escala$noite.2[is.na(escala$noite.2)]<-0# e na coluna noite.2
        falta.NA<-length(which(escala$noite!=0))#falta.NA guarda o
comprimento dos dias da coluna noite que nao tem ninguem para que garanta-se
que essa coluna esteja inteiramente preenchida

        if(falta.NA!=length(dias.do.mes))#se o comprimento de falta.NA for
diferente do numero de dias no mes, isso significa que tem "buracos" na
coluna a serem preenchidos
            {
                if(escala$noite[dias.3]==0)#para a condicao determinada
acima, se o dia sorteado na coluna noite tiver um 0
                {
                escala[dias.3,"noite"]=z# pode-se preencher com o
funcionario z
                } else {
                escala[dias.3,"noite.2"]=z#caso nao seja igual a 0, isso
significa que ja tem uma pessoa, entao a pessoa ira preencher a coluna
noite.2
                }
                keep[nrow(keep)-1,1]<-dias.3#esse dia sera guardado na
penultima linha da coluna z de keep (z = funcionario)
            }
        if(falta.NA==length(dias.do.mes))#caso o comprimento de falta.NA for
igual ao numero de dias no mes, entao nao precisara ocupar essa coluna,
ocupando entao a coluna noite.2
            {
                escala[dias.3,"noite.2"]=z#o funcionario z ira ocupar a
coluna noite.2 no dia sorteado em dias.3
                keep[nrow(keep)-2,1]<-dias.3#isso sera guardado na
antepenultima linha da coluna z da matriz keep
            }
        escala$noite[escala$noite=="0"]<-NA#a proxima etapa nao podera
correr com 0 nas colunas, pois ele conta os 0, o que nao é o objetivo do
```

```
comando
        escala$noite.2[escala$noite.2=="0"]<-NA#na linha acima e neste
linha, os 0 sao substituidos por NA's
        total.1<-table(unlist(subset(escala[,(2:4)])))#total.1 e o objeto
que ira guardar a contagem de ocorrencias de cada funcionarios nas colunas
dia,noite e noite.2
        numero.plantoes<-max(total.1)#esse objeto mostra qual e o numero
mais alto de total.1
        falta.preencher<-numero.plantoes-total.1#falta.preencher e o objeto
que calcula quantos plantoes faltam para cada funcionario para chegar ao
numero maximo de plantao calculado no objeto anterior
        teste.total<-cbind(total.1,falta.preencher)#teste.total une em uma
tabela o objeto total.1 e o objeto falta.preencher
        }
    }

    escala$noite[is.na(escala$noite)]<-0#o ciclo anterior finalizou com os
0's sendo NAs. Para a proxima etapa, ha a necessidade de converter novamente
para 0
    escala$noite.2[is.na(escala$noite.2)]<-0#o mesmo qeu acima para a coluna
noite.2
#a partir daqui, o objetivo é preencher os "buracos" da coluna noite que
ainda restaram
    onde.falta.na.coluna.noite<-which(escala$noite==0)#esse objeto ira
guardar as posicoes da coluna noite que ainda faltam pessoas (onde tem 0)
    onde.tem.pessoa.na.coluna.noite.2<-which(escala$noite.2!=0)#esse objeto
guarda os dias que tem pessoa na coluna noite.2 que pode ser passada para a
coluna noite

    while(length(onde.falta.na.coluna.noite)!=0)#enquanto o comprimento de
onde.falta.na.coluna.noite for diferente de 0, é necessario colocar pessoas
nas posicoes que tem 0
    {
        for(m in 1:length(onde.falta.na.coluna.noite))#inicia-se o ciclo
para cada dia m que falta funcionario na coluna noite
        {
        for(k in 1:length(onde.tem.pessoa.na.coluna.noite.2))#para cada
pessoa k que esta na coluna noite.2
        {
            guarda.posicao.dia.que.vai.sair<-
onde.tem.pessoa.na.coluna.noite.2[k]#esse objeto guarda a posicao do dia que
a pessoa ira sair da coluna noite.2
            guarda.pessoa<-
escala$noite.2[guarda.posicao.dia.que.vai.sair]#esse objeto guarda a pessoa
que esta na posicao do dia que ira ser tirado da coluna noite.2
            guarda.dia.novo<-onde.falta.na.coluna.noite[m]#esse objeto
guarda o dai que essa pessoa sera inserida
            verifica.dia.em.descanso<-
guarda.dia.novo%in%descanso[guarda.pessoa,]#esse objeto verifica se a pessoa
selecionada da coluna noite.2 pode trabalhar no dia que falta alguem na
```

```
coluna noite por meio da tabela descanso
            if(verifica.dia.em.descanso)#se o objeto
verifica.dia.em.descanso for TRUE, isso significa que a pessoa podera
trabalhar naquele dia da coluna noite que esta com 0
            {
                escala$noite[guarda.dia.novo]<-guarda.pessoa#dessa forma,
essa pessoa sera colocada nesse novo dia da coluna noite
                escala$noite.2[guarda.posicao.dia.que.vai.sair]<-0#e sera
adicionado um 0 no lugar que ela ocupava na coluna noite.2
                break#dessa forma,nao sera mais preciso procurar alguma
pessoa para ocupar esse dia, podendo passar para o proximo m
            }
        }
        }
    onde.falta.na.coluna.noite<-which(escala$noite==0)#esse objeto ira
guardar as posicoes da coluna noite que ainda faltam pessoas (onde tem 0)
    onde.tem.pessoa.na.coluna.noite.2<-which(escala$noite.2!=0)#esse objeto
guarda os dias que tem pessoa na coluna noite.2 que pode ser passada para a
coluna noite
    }
}

escala$noite[escala$noite=="0"]<-NA#novamente serao convertidos os 0's em
NA's para calcular os plantoes
escala$noite.2[escala$noite.2=="0"]<-NA#o mesmo que acima

total.1<-table(unlist(subset(escala[,(2:4)])))#total.1 e o objeto que ira
guardar a contagem de ocorrencias de cada funcionarios nas colunas dia,noite
e noite.2
numero.plantoes<-max(total.1)#esse objeto mostra qual e o numero mais alto
de total.1
falta.preencher<-numero.plantoes-total.1#falta.preencher e o objeto que
calcula quantos plantoes faltam para cada funcionario para chegar ao numero
maximo de plantao calculado no objeto anterior
teste.total<-cbind(total.1,falta.preencher)#teste.total une em uma tabela o
objeto total.1 e o objeto falta.preencher

quantos.dia<-table(escala$dia)#conta o numero de plantoes diurnos de cada
funcionario
quantos.noite<-table(unlist(subset(escala[,(3:4)])))#conta o numero de
plantoes noturnos de cada funcionario
quantos.dia<-as.data.frame(quantos.dia)#converte table para data frame
quantos.noite<-as.data.frame(quantos.noite)#converte table para data frame
contagem<-data.frame(teste.total[,1],quantos.dia[,2],quantos.noite[,2])#gera
um data frame contagem com o total de plantoes, diurnos e noturnos
colnames(contagem)<-c("Plantões(Total)","Diurnos(Num)","Noturnos(Num)")#muda
o nome das colunas

print(contagem)#mostra a contagem de numero de plantoes de cada funcionario,
total, diurno e noturno
```

```
escala$noite[escala$noite=="0"]<-" "#substitui os 0's por espaços para que
nao apareceram na planilha final
escala$noite.2[escala$noite.2=="0"]<-" "#o mesmo que acima

escala<-data.frame(semana,escala)#une a escala gerada com os dias da semana
escala$semana[escala$semana=="1"]<-"Domingo"#substitui os dias da semana de
1 a 7, sendo 1 = domingo
escala$semana[escala$semana=="2"]<-"Segunda"#2=segunda
escala$semana[escala$semana=="3"]<-"Terça"#3=terca
escala$semana[escala$semana=="4"]<-"Quarta"#4=quarta
escala$semana[escala$semana=="5"]<-"Quinta"#5=quinta
escala$semana[escala$semana=="6"]<-"Sexta"#6=sexta
escala$semana[escala$semana=="7"]<-"Sábado"#7=sabado

colnames(escala)<-c("Semana","Dia","Diurno","Noturno 1","Noturno 2")#muda os
nomes das colunas da escala final

print(escala)#gera data frame com a escala final
}
```