

Álvaro de Prada Martínez

Programación en R

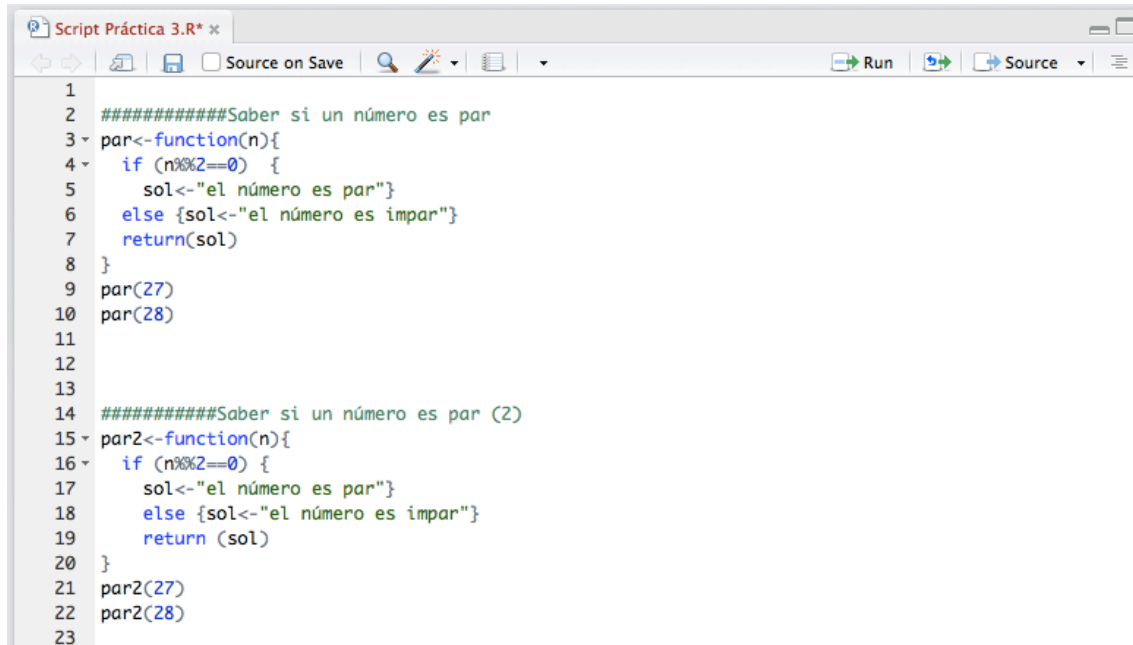
Práctica 3

Programación con R. Práctica 3

Ejercicio 1. Programación de Scripts y Funciones.

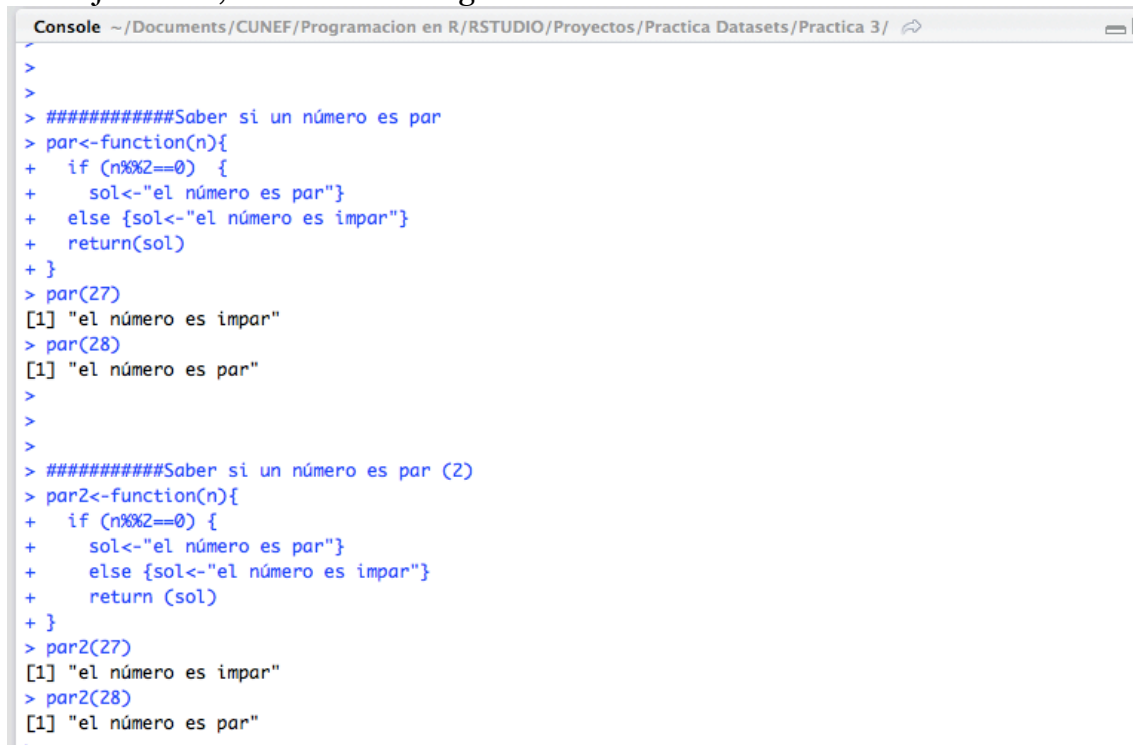
1.1 Saber si un número es par:

Ejecutamos el script siguiente, el cual comprobará de dos formas diferentes si un número es par o si es impar:



```
1
2 #####Saber si un número es par
3 par<-function(n){
4   if (n%%2==0) {
5     sol<-"el número es par"}
6   else {sol<-"el número es impar"}
7   return(sol)
8 }
9 par(27)
10 par(28)
11
12
13
14 #####Saber si un número es par (2)
15 par2<-function(n){
16   if (n%%2==0) {
17     sol<-"el número es par"}
18   else {sol<-"el número es impar"}
19   return (sol)
20 }
21 par2(27)
22 par2(28)
23
```

Tras ejecutarlo, se devuelve el siguiente resultado:



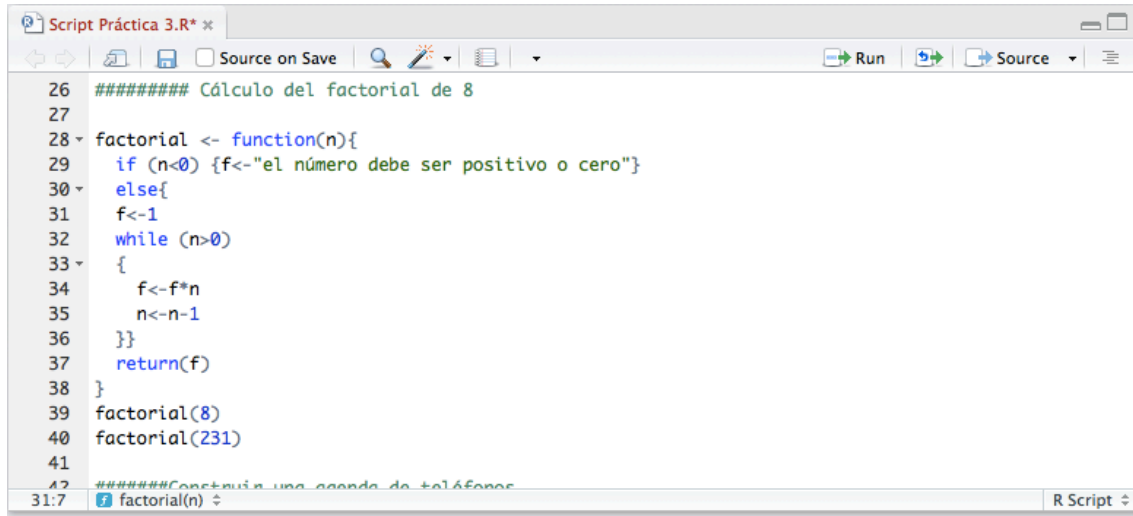
```
Console ~/Documents/CUNEF/Programacion en R/RSTUDIO/Proyectos/Practica Datasets/Practica 3/
>
>
> #####Saber si un número es par
> par<-function(n){
+   if (n%%2==0) {
+     sol<-"el número es par"}
+   else {sol<-"el número es impar"}
+   return(sol)
+ }
> par(27)
[1] "el número es impar"
> par(28)
[1] "el número es par"
>
>
>
> #####Saber si un número es par (2)
> par2<-function(n){
+   if (n%%2==0) {
+     sol<-"el número es par"}
+   else {sol<-"el número es impar"}
+   return (sol)
+ }
> par2(27)
[1] "el número es impar"
> par2(28)
[1] "el número es par"
```

Álvaro de Prada Martínez

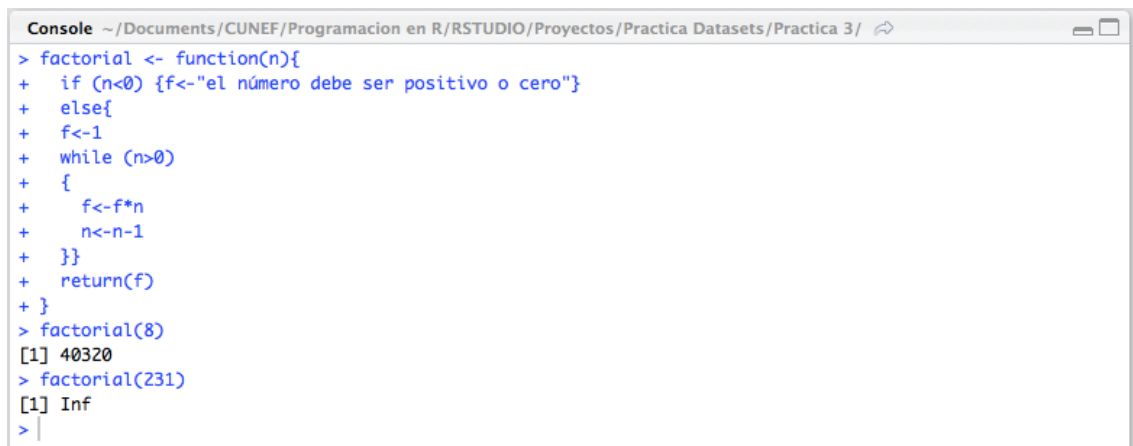
Programación en R

Práctica 3

1.2 Calcular el factorial de 8 y calcular el factorial de 231:



```
26 ##### Cálculo del factorial de 8
27
28 factorial <- function(n){
29   if (n<0) {f<-"el número debe ser positivo o cero"}
30   else{
31     f<-1
32     while (n>0)
33     {
34       f<-f*n
35       n<-n-1
36     }}
37   return(f)
38 }
39 factorial(8)
40 factorial(231)
41
42 ##### Construir una agenda de teléfonos
31:7 factorial(n) ↕
```



```
Console ~/Documents/CUNEF/Programacion en R/RSTUDIO/Proyectos/Practica Datasets/Practica 3/ ↗
> factorial <- function(n){
+   if (n<0) {f<-"el número debe ser positivo o cero"}
+   else{
+     f<-1
+     while (n>0)
+     {
+       f<-f*n
+       n<-n-1
+     }}
+   return(f)
+ }
> factorial(8)
[1] 40320
> factorial(231)
[1] Inf
> |
```

*Factorial de 231: El factorial de 231 lo devuelve como infinito. A partir de 271 considera infinito los factoriales al no poder mostrarlos en pantalla debido a su longitud.

*1.3 Construir un vector añadiendo elementos por la derecha:
Crear una agenda con nombres y teléfonos de tres amigos.*

```
Script Práctica 3.R* x
Source on Save
Run
Source

108 #####
109 #####Construir una agenda de teléfonos#####
110 nueva.agenda<-function(){
111   ##esta función crea una agenda
112   ##que almacenar en un dataframe con dos campos: nombres, telefonos
113   nombres<-"Victoria"
114   telefonos<-"234432234"
115   continuar<-"S";
116   while (continuar == "S") {
117     nombre<-readline(prompt="Dime el nombre: ")
118     telefono<-readline(prompt="Dime el teléfono: ")
119     nombres<-c(nombres,nombre)
120     telefonos<-c(telefonos,telefono)
121     continuar <- readline(prompt="¿Continuar(S/N)? ")
122   }
123   mdf<-data.frame(nombres, telefonos)
124   mdf$nombres<-as.character(mdf$nombres)
125   mdf$telefonos<-as.character(mdf$telefonos)
126   return (mdf)
127 }
128 miagenda<-nueva.agenda()
129
```

128:25 Construir una agenda de teléfonos R Script

```
Console ~/Documents/CUNEF/Programacion en R/RSTUDIO/Proyectos/Practica Datasets/Practica 3/
> #####Construir una agenda de teléfonos#####
> nueva.agenda<-function(){
+   ##esta función crea una agenda
+   ##que almacenar en un dataframe con dos campos: nombres, telefonos
+   nombres<-"Victoria"
+   telefonos<-"234432234"
+   continuar<-"S";
+   while (continuar == "S") {
+     nombre<-readline(prompt="Dime el nombre: ")
+     telefono<-readline(prompt="Dime el teléfono: ")
+     nombres<-c(nombres,nombre)
+     telefonos<-c(telefonos,telefono)
+     continuar <- readline(prompt="¿Continuar(S/N)? ")
+   }
+   mdf<-data.frame(nombres, telefonos)
+   mdf$nombres<-as.character(mdf$nombres)
+   mdf$telefonos<-as.character(mdf$telefonos)
+   return (mdf)
+ }
> miagenda<-nueva.agenda()
Dime el nombre: Alvaro
Dime el teléfono: 647329523
¿Continuar(S/N)? S
Dime el nombre: Sebas
Dime el teléfono: 888444999
¿Continuar(S/N)? S
Dime el nombre: Sergio
Dime el teléfono: 666666666
¿Continuar(S/N)? N
>
```

Álvaro de Prada Martínez
Programación en R
Práctica 3

	nombres	telefonos
1	Victoria	234432234
2	Alvaro	647329523
3	Sebas	888444999
4	Sergio	666666666

1.4 Añadir mediante `add.agenda` a dos amigos más a la anterior agenda.

```
Script Práctica 3.R* x miagenda x
Filter
nombres telefonos
1 Victoria 234432234
2 Alvaro 647329523
3 Sebas 888444999
4 Sergio 666666666
```

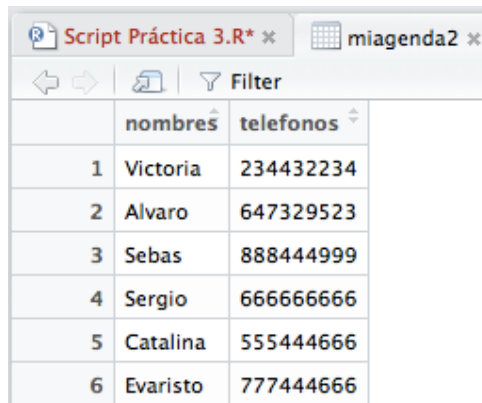
```
131 #####Construir una agenda de teléfonos#####
132 add.agenda<-function(agenda.ant){
133   ###esta función añade nombres y teléfonos a una agenda
134   ###que está almacenada en un dataframe con dos campos: nombres, telefonos
135   nombres<-agenda.ant$nombres
136   telefonos<-agenda.ant$telefonos
137   continuar<-"S";
138   while (continuar == "S") {
139     nombre<-readline(prompt="Dime el nombre: ")
140     telefono<-readline(prompt="Dime el teléfono: ")
141     nombres<-c(nombres,nombre)
142     telefonos<-c(telefonos,telefono)
143     continuar <- readline(prompt="¿Continuar(S/N)? ")
144   }
145   return (data.frame(nombres, telefonos))
146 }
147
148 miagenda2<-add.agenda(miagenda)
149
```

```
137:18 add.agenda(agenda.ant) R Script
```

```
Console ~/Documents/CUNEF/Programacion en R/RSTUDIO/Proyectos/Practica Datasets/Practica 3/
> #####Construir una agenda de teléfonos#####
> add.agenda<-function(agenda.ant){
+   ###esta función añade nombres y teléfonos a una agenda
+   ###que está almacenada en un dataframe con dos campos: nombres, telefonos
+   nombres<-agenda.ant$nombres
+   telefonos<-agenda.ant$telefonos
+   continuar<-"S";
+   while (continuar == "S") {
+     nombre<-readline(prompt="Dime el nombre: ")
+     telefono<-readline(prompt="Dime el teléfono: ")
+     nombres<-c(nombres,nombre)
+     telefonos<-c(telefonos,telefono)
+     continuar <- readline(prompt="¿Continuar(S/N)? ")
+   }
+   return (data.frame(nombres, telefonos))
+ }
>
> miagenda2<-add.agenda(miagenda)
Dime el nombre: Catalina
Dime el teléfono: 555444666
¿Continuar(S/N)? S
Dime el nombre: Evaristo
Dime el teléfono: 777444666
¿Continuar(S/N)? N
>
```

Álvaro de Prada Martínez
Programación en R
Práctica 3

Resultado:



The screenshot shows an RStudio window with two tabs: 'Script Práctica 3.R*' and 'miagenda2'. The 'miagenda2' tab is active, displaying a data table with 6 rows and 3 columns. The columns are labeled 'nombres' and 'telefonos'. The first column contains indices from 1 to 6. The data is as follows:

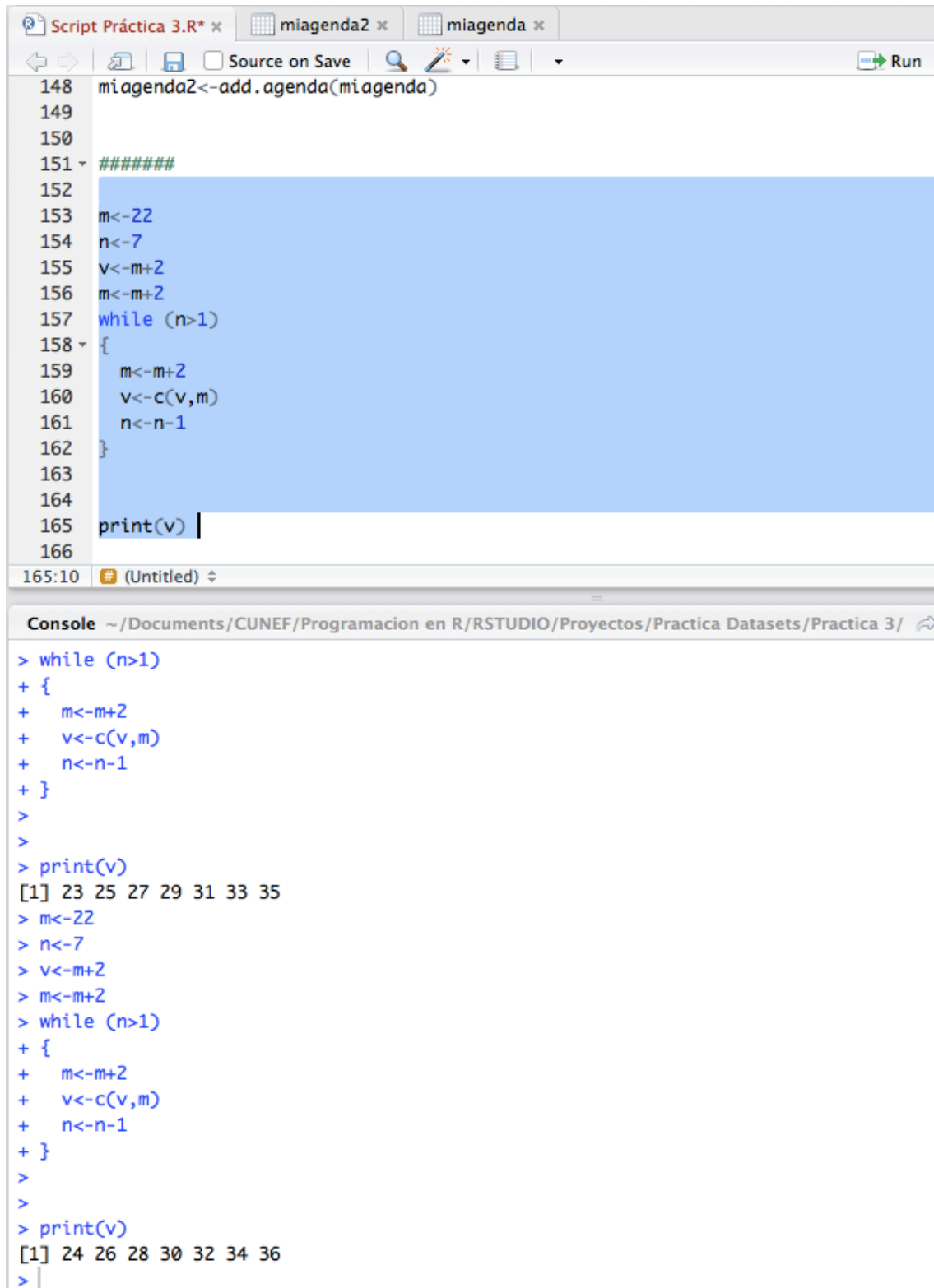
	nombres	telefonos
1	Victoria	234432234
2	Alvaro	647329523
3	Sebas	888444999
4	Sergio	666666666
5	Catalina	555444666
6	Evaristo	777444666

Álvaro de Prada Martínez

Programación en R

Práctica 3

1.5 Construir un vector que contenga los n números pares siguientes a m (par):



The screenshot shows the RStudio interface. The top pane displays the R script 'Script Práctica 3.R' with the following code:

```
148 miagenda2<-add.agenda(miagenda)
149
150
151 #####
152
153 m<-22
154 n<-7
155 v<-m+2
156 m<-m+2
157 while (n>1)
158 {
159   m<-m+2
160   v<-c(v,m)
161   n<-n-1
162 }
163
164
165 print(v)
166
```

The bottom pane shows the console output, which includes the code being executed and the resulting vector values:

```
> while (n>1)
+ {
+   m<-m+2
+   v<-c(v,m)
+   n<-n-1
+ }
>
>
> print(v)
[1] 23 25 27 29 31 33 35
> m<-22
> n<-7
> v<-m+2
> m<-m+2
> while (n>1)
+ {
+   m<-m+2
+   v<-c(v,m)
+   n<-n-1
+ }
>
>
> print(v)
[1] 24 26 28 30 32 34 36
>
```

Álvaro de Prada Martínez

Programación en R

Práctica 3

Ejercicio 2. Realizar las modificaciones pertinentes en el último ejemplo para que también funcione si el número m es impar, almacenar todo en un nuevo script y ejecutarlo para ver su funcionamiento. Para ello usar 'Interpretar código fuente' del menú de consola o ejecutar el comando: `source("ruta/fichero.R")`:



The screenshot shows the RStudio interface. The top pane displays an R script with the following code:

```
1 m<-21
2 n<-7
3 v<-m+2
4 m<-m+2
5 if (m%%2!=0){m<-m+1}
6 while (n>1)
7 {
8   m<-m+2
9   v<-c(v,m)
10  n<-n-1
11 }
12
13
14 print(v)
```

The bottom pane shows the console output, which includes the execution of the script and subsequent manual commands:

```
+ n<-n-1
+ }
>
>
> print(v)
[1] 23 25 27 29 31 33 35
> m<-22
> n<-7
> v<-m+2
> m<-m+2
> while (n>1)
+ {
+   m<-m+2
+   v<-c(v,m)
+   n<-n-1
+ }
>
>
> print(v)
[1] 24 26 28 30 32 34 36
> source('~/.active-rstudio-document')
[1] 23 26 28 30 32 34 36
> source('~/.active-rstudio-document')
[1] 23 26 28 30 32 34 36
>
```