Práctica IV: Respuesta en frecuencia de sistemas LTI

Señales y Sistemas, Grado en Ciencia de Datos - UV

Sandra Paniagua Sánchez y Gema Bravo Aguilera

2022-11-25

Contents

Práctica IV: Respuesta en frecuencia de sistemas LTI

1. Objetivos

En esta práctica se estudia la respuesta en frecuencia de los sistemas LTI (discretos, los continuos se muestran en un Anexo que no forma parte de la práctica). Veremos cómo la podemos utilizar para determinar la salida en el estado estacionario de un sistema dada una entrada. Se analizarán sistemas con una determinada función de transferencia y, por otra parte, se diseñarán sistemas para obtener una respuesta en frecuencia.

2. Prelaboratorio:

Explica la relación entre la transformada de Fourier y la transformada Z

La transformada Z es una generalización de la transformada de Fourier de t discreto. Puede hallarse evaluando la transdormada Z $X(z) = e^{jw}$, o lo que es lo mismo, evaluando en el círculo unidad.

2.1. Sistemas discretos

En este apartado estudiaremos la relación que hay entre la posición de los polos y ceros de una transformada Z de un determinado sistema y su respuesta en frecuencia. Para ello consideraremos funciones de transferencia de la forma:

$$H_1(z) = 1 - 2 \cdot r \cdot cos(w_0) \cdot z^{-1} + r^2 \cdot z^{-2}$$
$$H_2(z) = \frac{1}{H_1(z)}$$

¿Cómo se relacionan r y w_0 con los ceros y polos en el dominio de la transformada Z (`zplane`)? Determina la respuesta impulsional (`impz`) y la respuesta en frecuencia (`freqz`) para diferentes valores de r (mayores, cercanos y menores que la unidad) y w_0. ¿Qué efectos tienen estos parámetros sobre la respuesta en frecuencia?

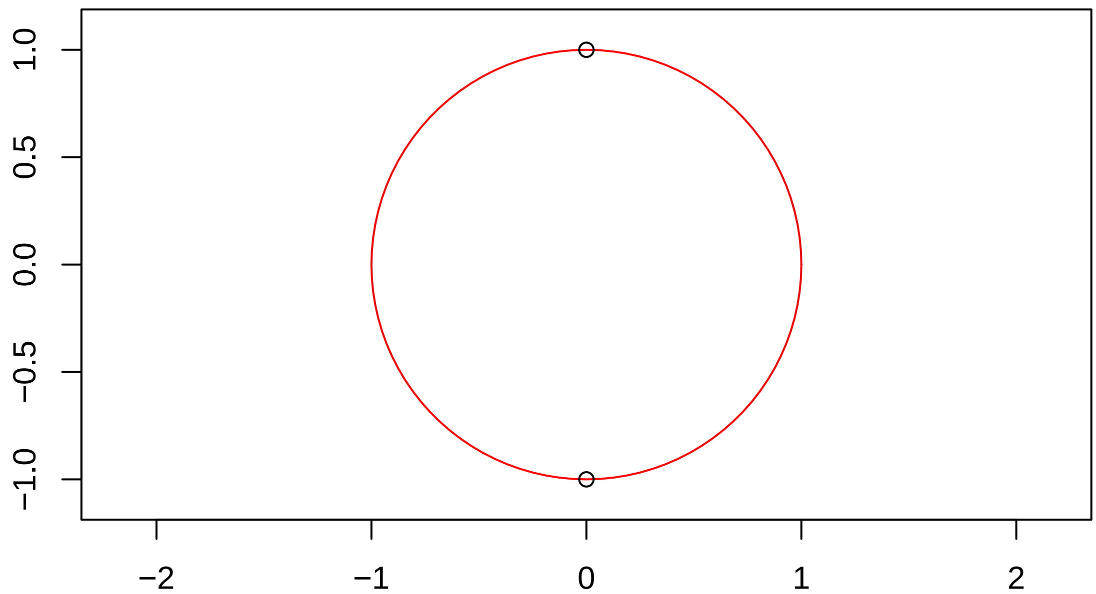
Paso 1 - Determinar los vectores de cada entrada:

Para H_1: num1 = c(1, $-2 \cdot r \cdot cos(w_0)$, r^2) den1 = c(1)

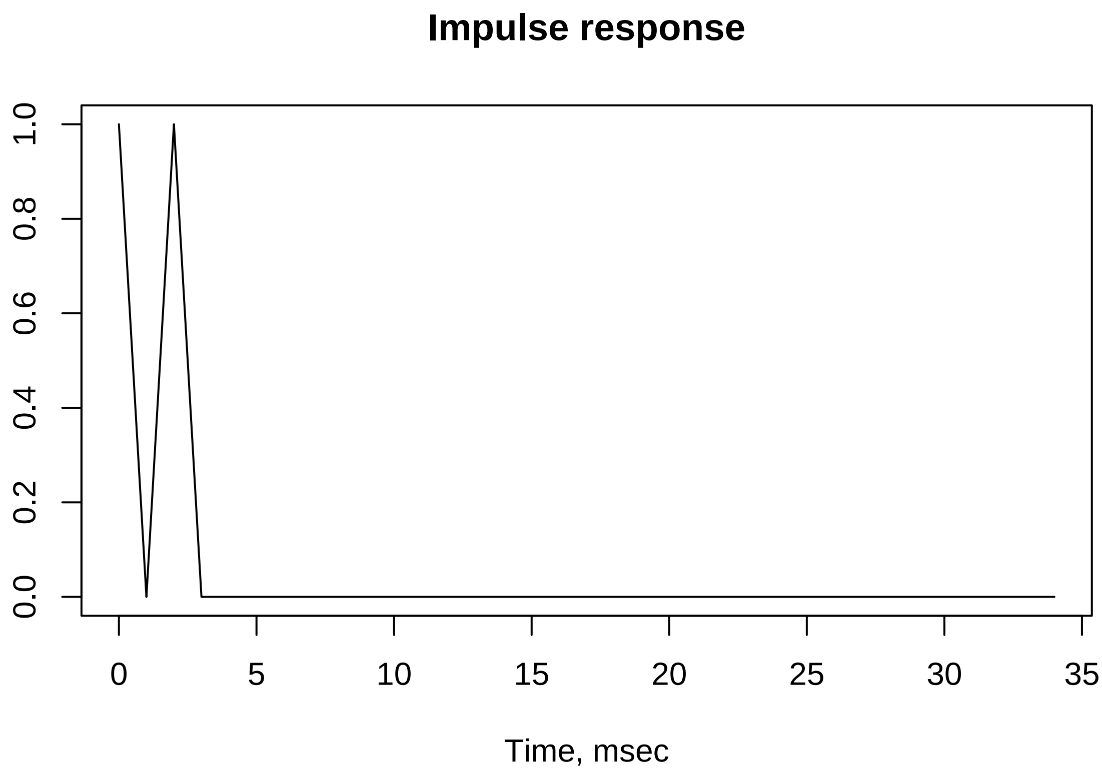
Para H_2: num2 = c(1) den2 = c(1, $-2 \cdot r \cdot cos(w_0)$, r^2)

Y ahora le daremos valores a w_0 con un r fijo:

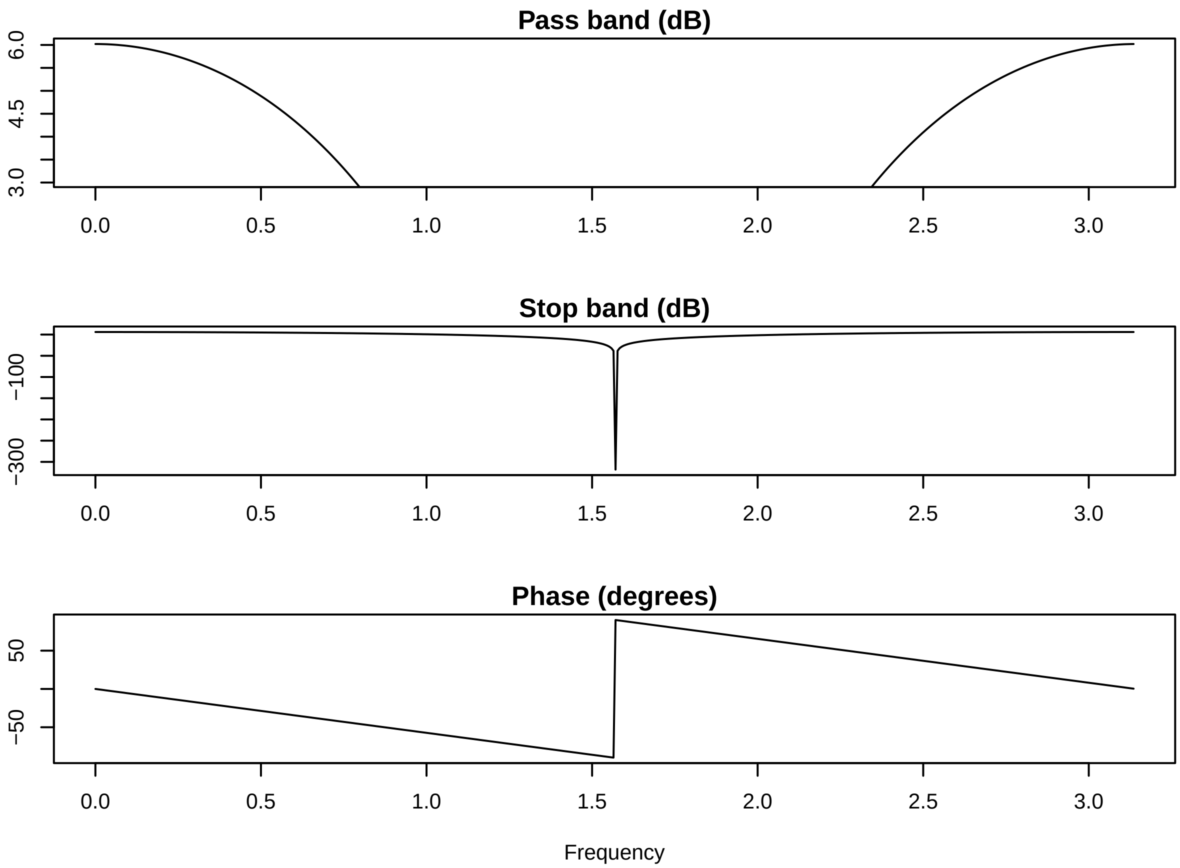
```
#Para H1:

r = 1
w0 = pi /2

num = c(1, -2*r*cos(w0), r^2)
den = c(1)

zplane(num,den)
```



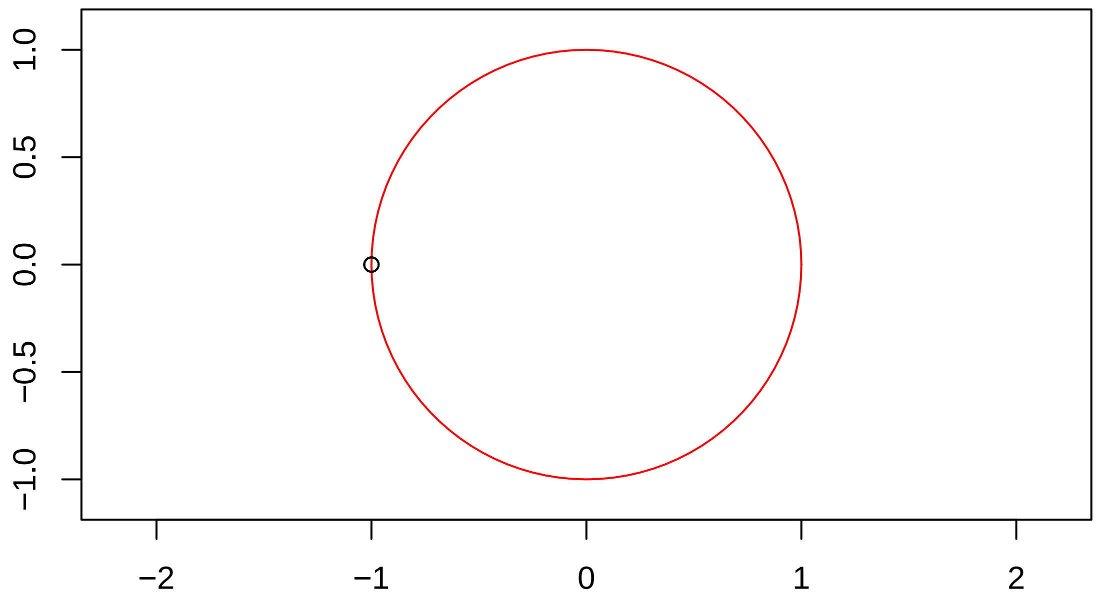

```
impz(num,den, 35)
```



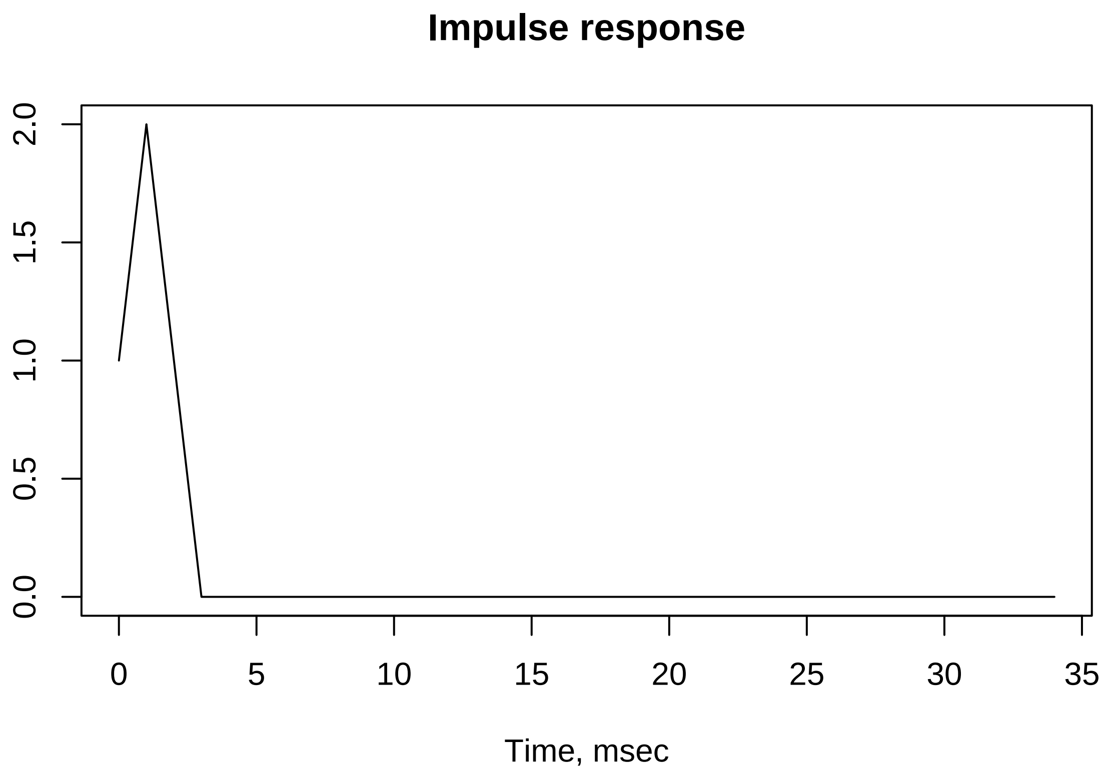

```
freqz(num,den)
```



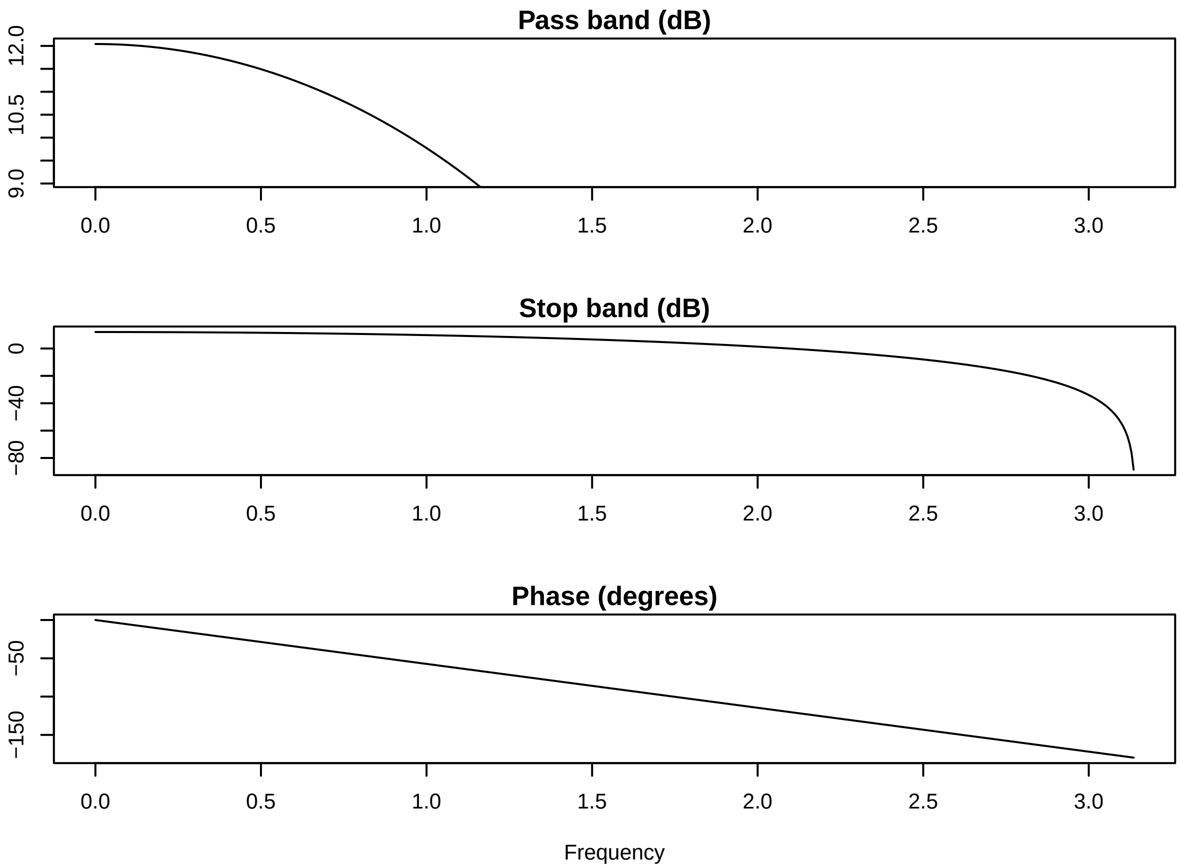

```r
r = 1
w0 = pi

num = c(1, -2*r*cos(w0), r^2)
den = c(1)

zplane(num,den)
```



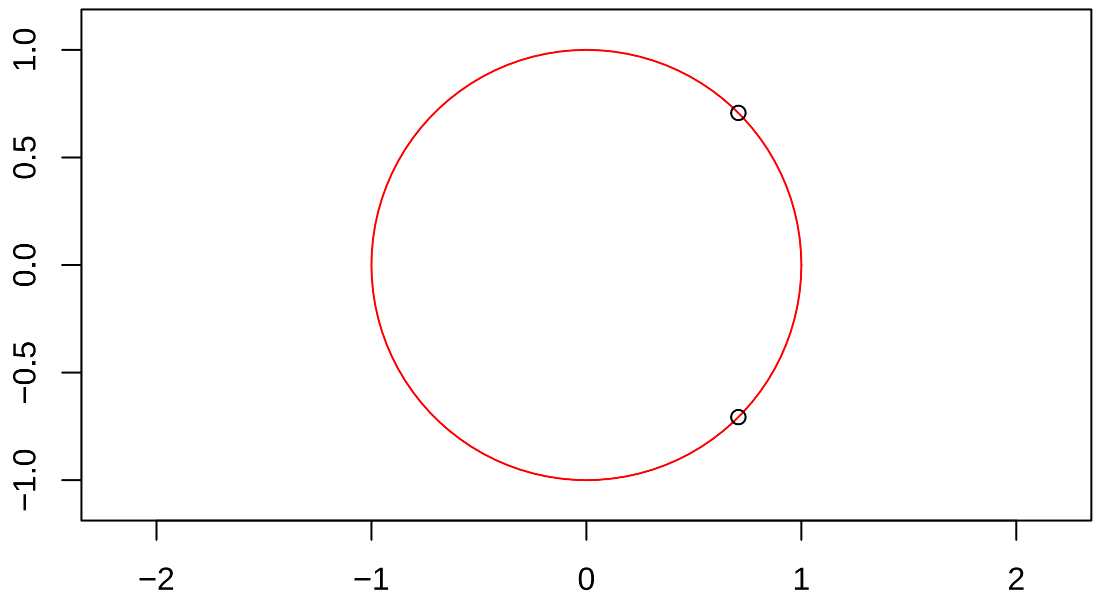

```
impz(num,den, 35)
```



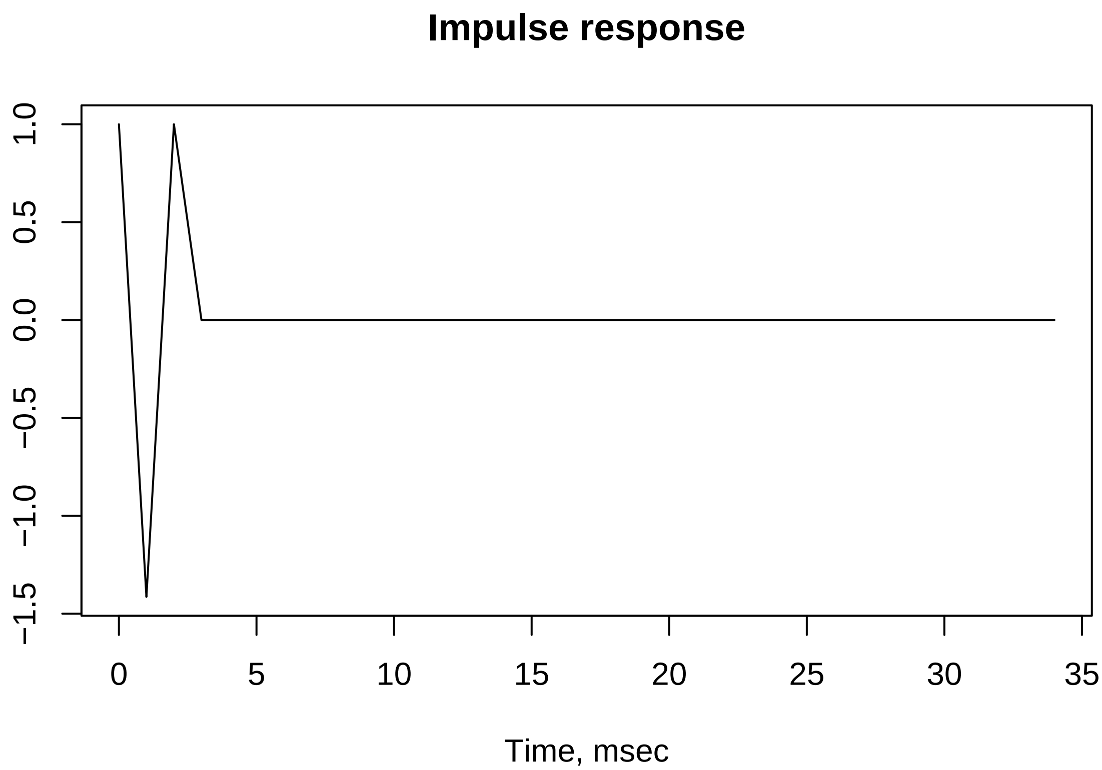

```
freqz(num,den)
```



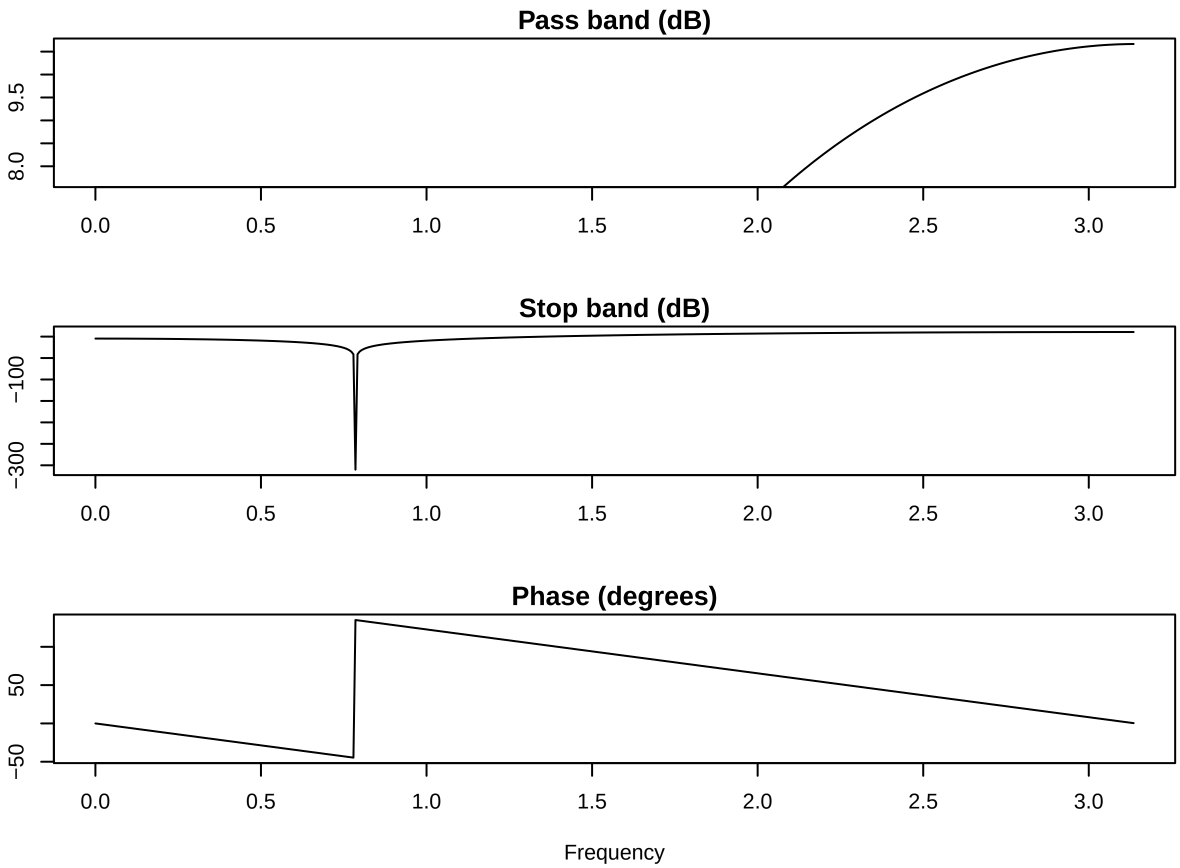

```
r = 1
w0 = pi /4

num = c(1, -2*r*cos(w0), r^2)
den = c(1)

zplane(num,den)
```



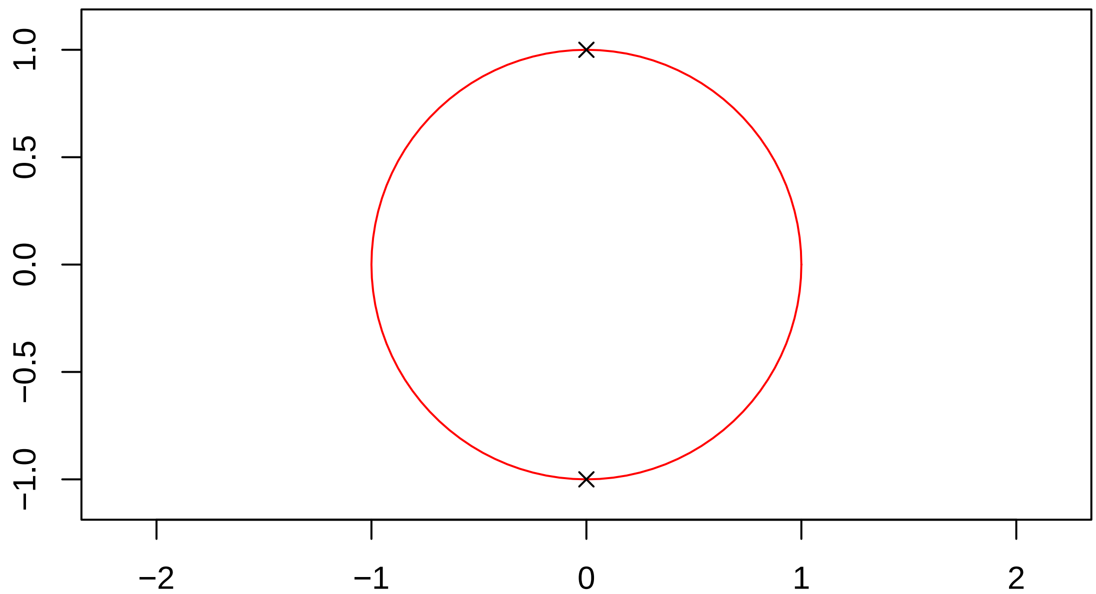

```
impz(num,den, 35)
```



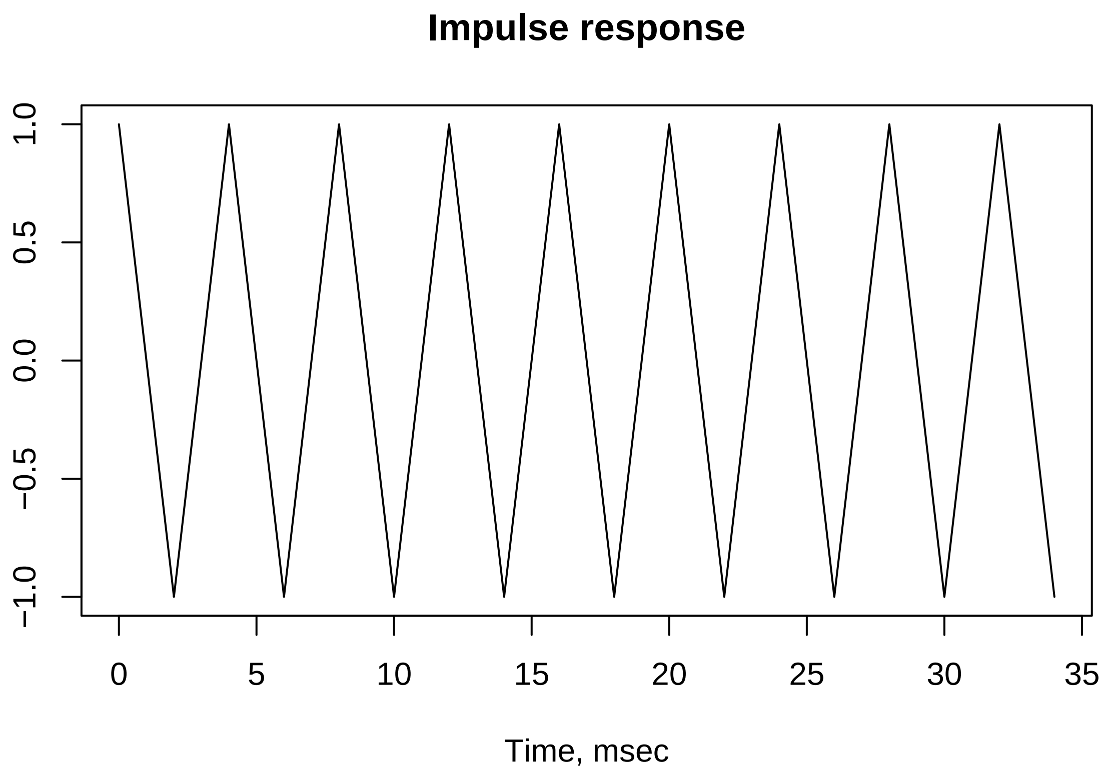

```
freqz(num,den)
```



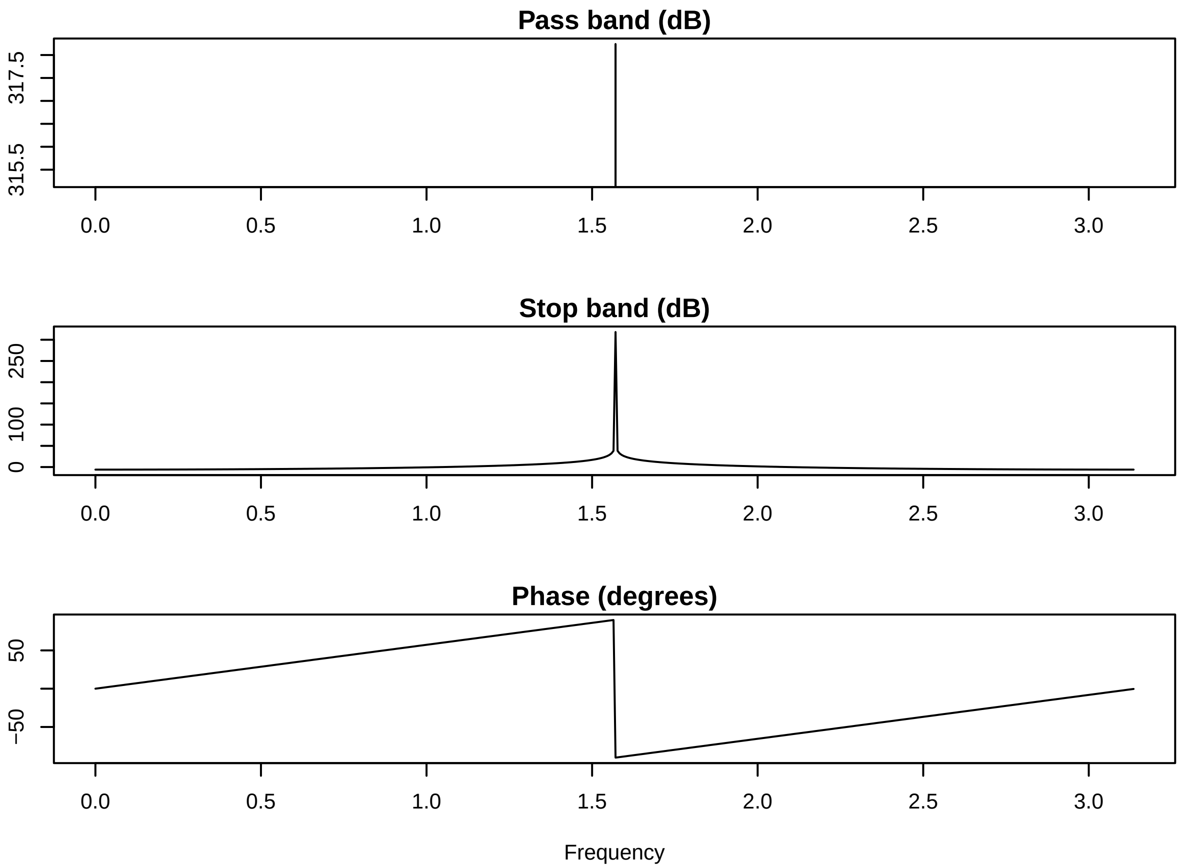

```
#Para H2:

r = 1
w0 = pi /2

num = c(1)
den = c(1, -2*r*cos(w0), r^2)

zplane(num,den)
```



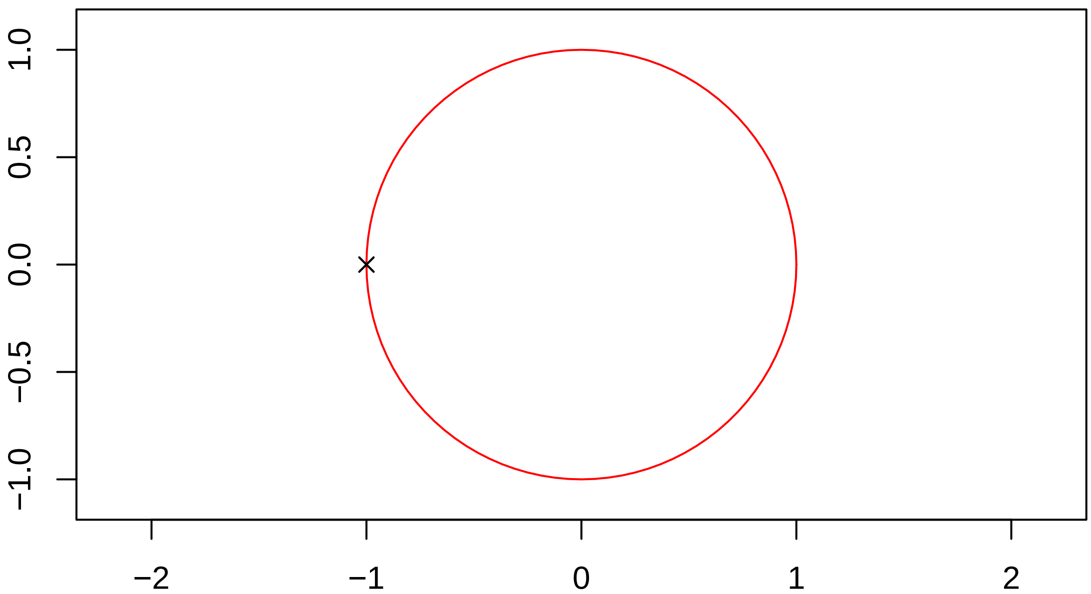

```
impz(num,den, 35)
```



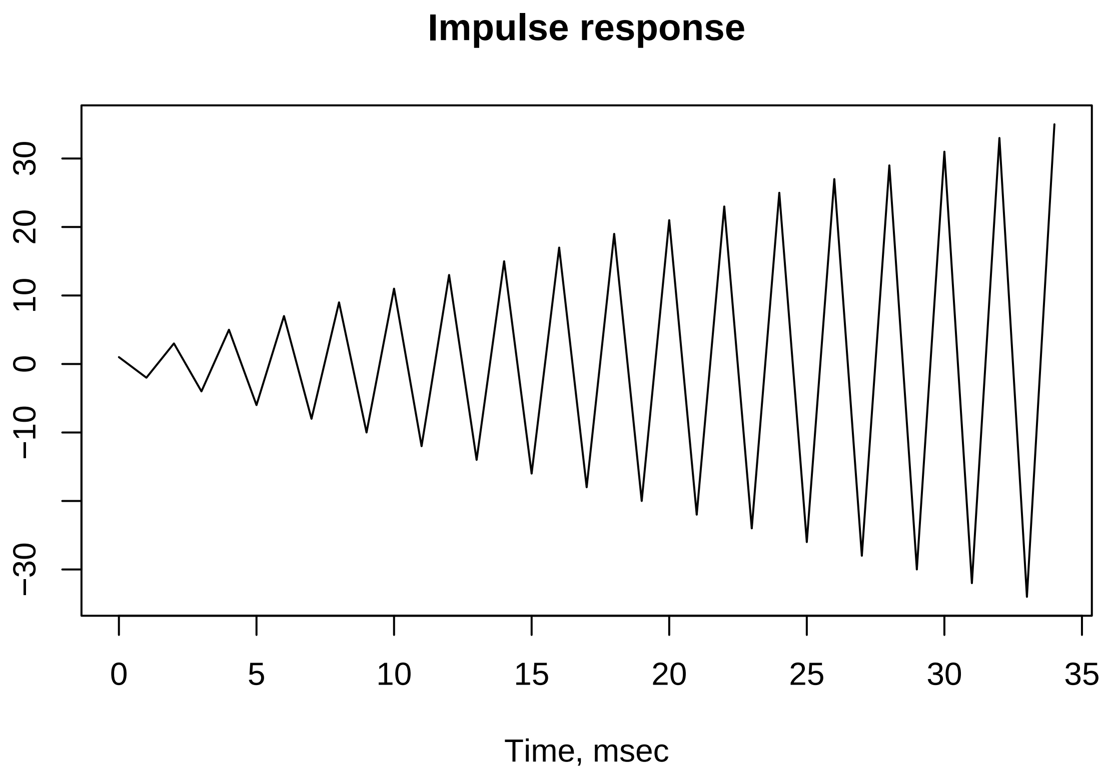

```
freqz(num,den)
```



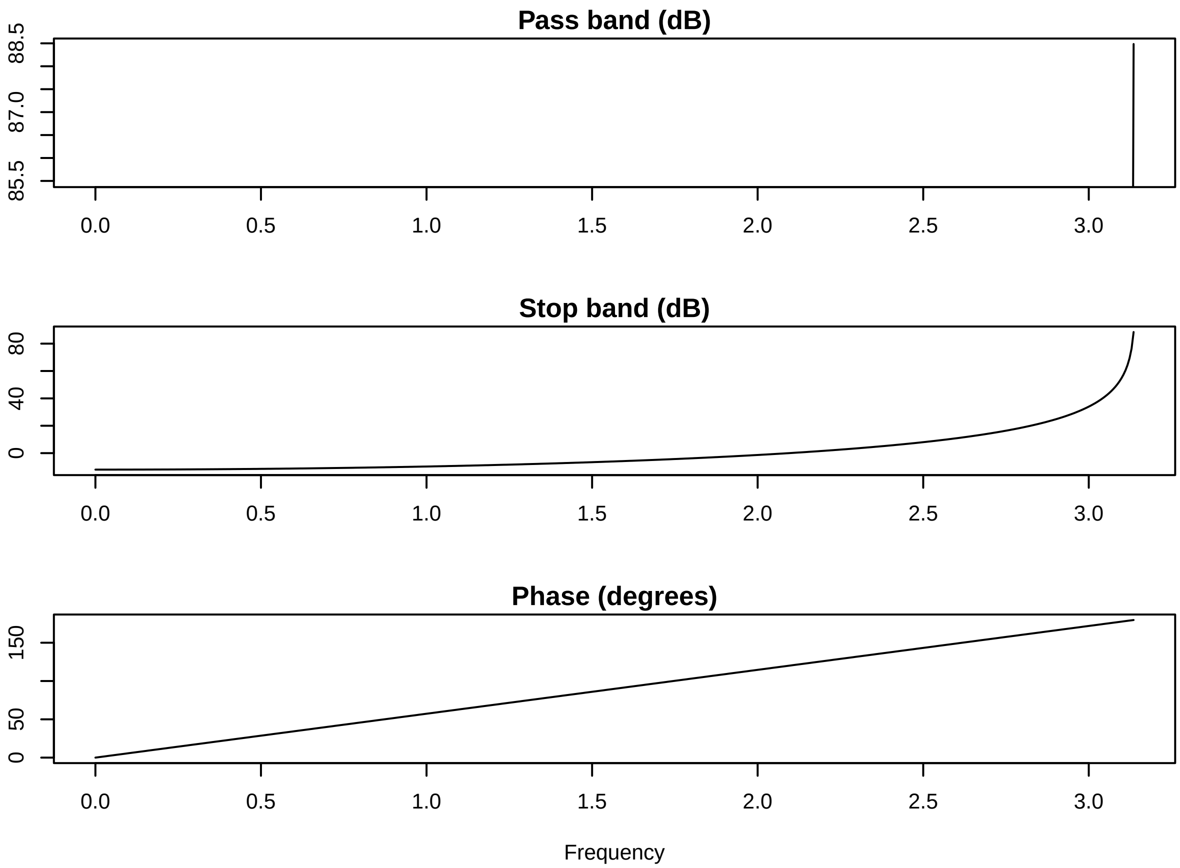

```
r = 1
w0 = pi

num = c(1)
den = c(1, -2*r*cos(w0), r^2)

zplane(num,den)
```



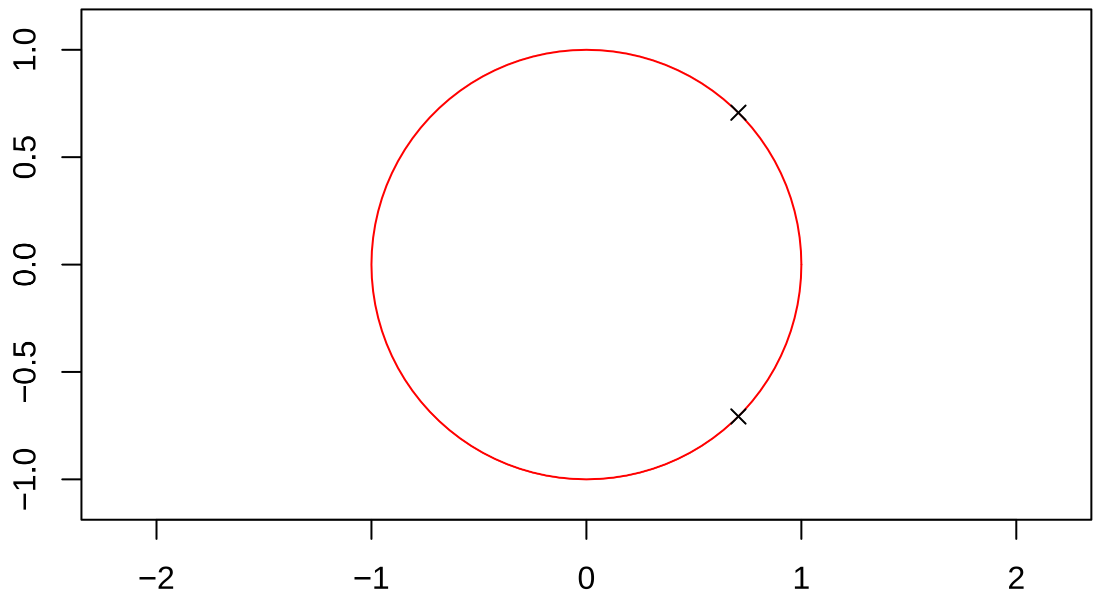

```
impz(num,den, 35)
```



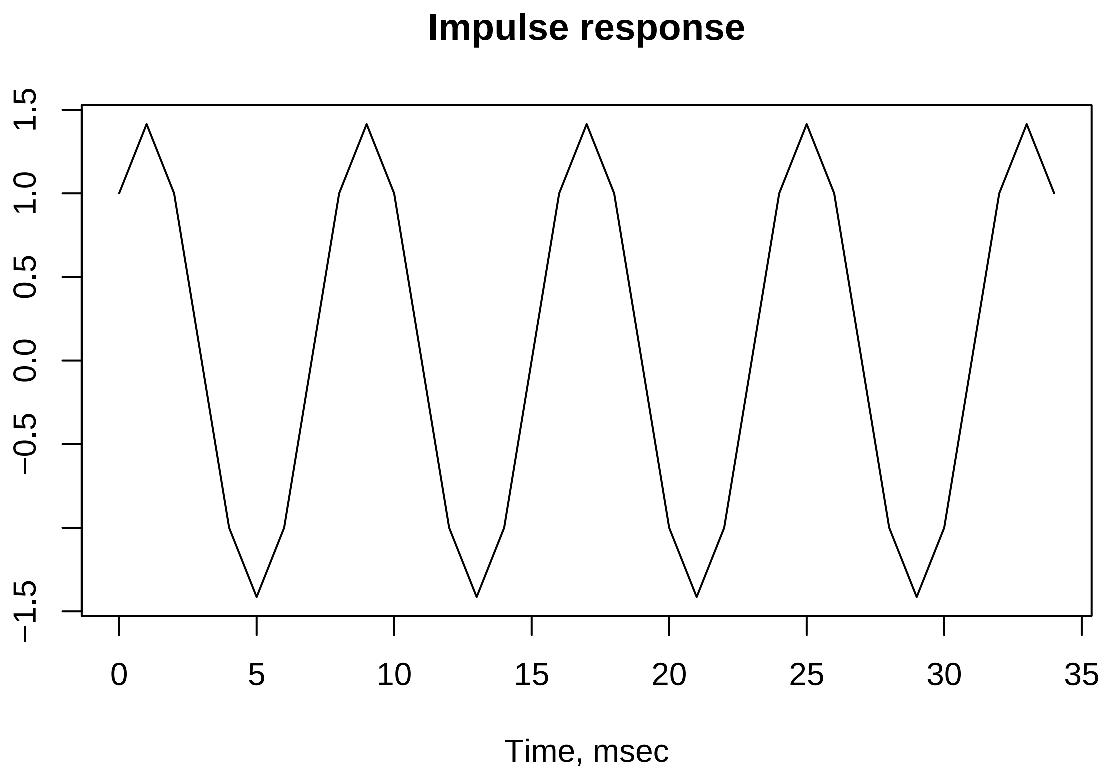

```
freqz(num,den)
```



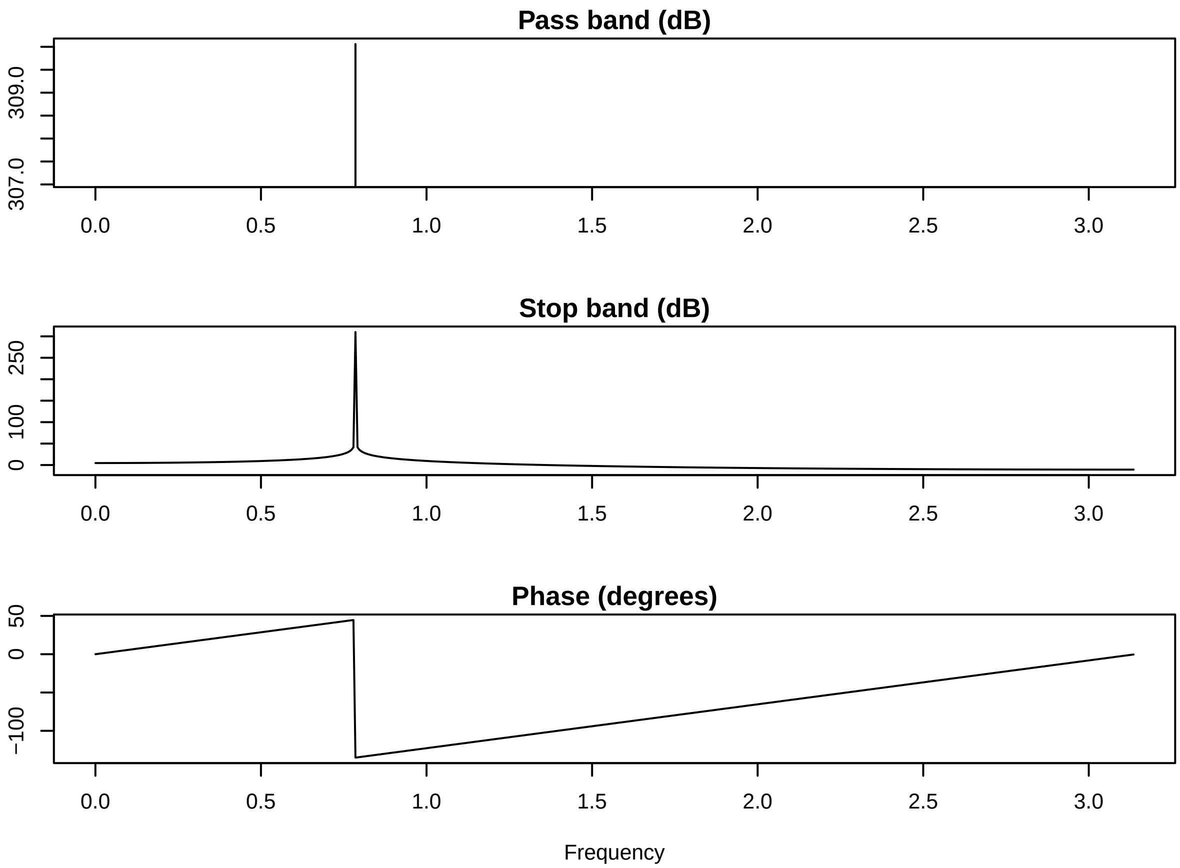

```
r = 1
w0 = pi /4

num = c(1)
den = c(1, -2*r*cos(w0), r^2)

zplane(num,den)
```




```
impz(num,den, 35)
```

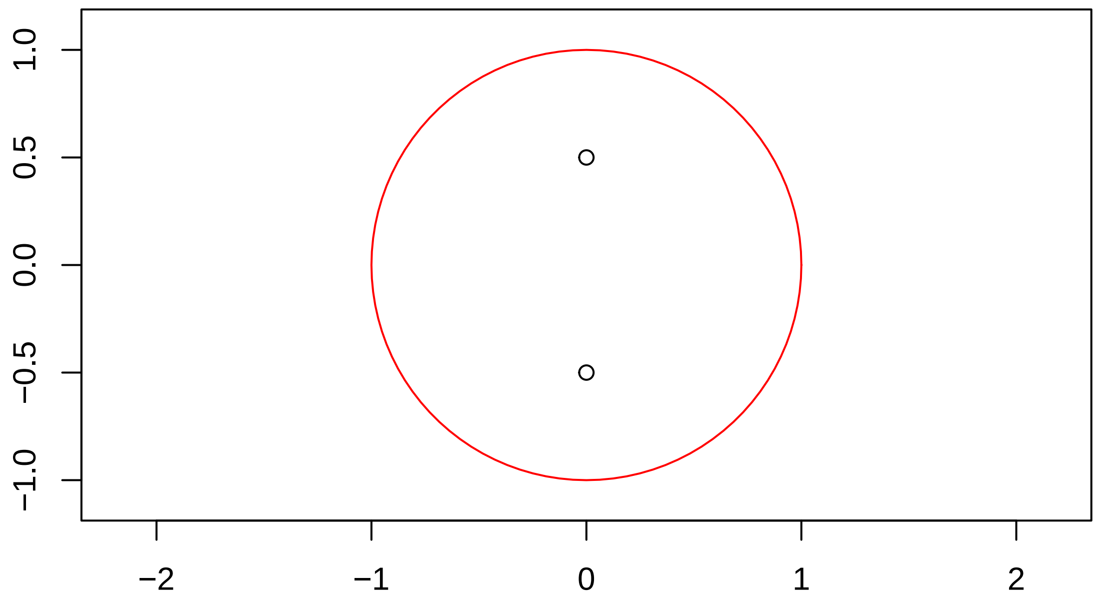



```
freqz(num,den)
```

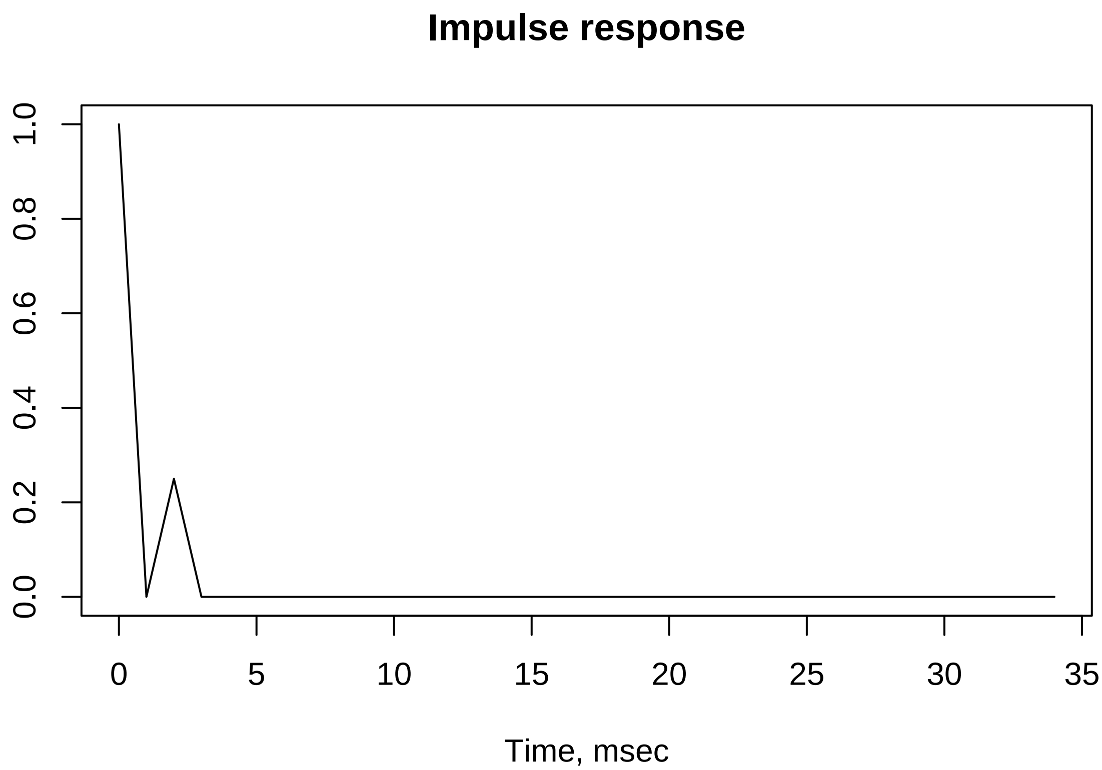


Y ahora le daremos valores a r con un w_0 fijo:

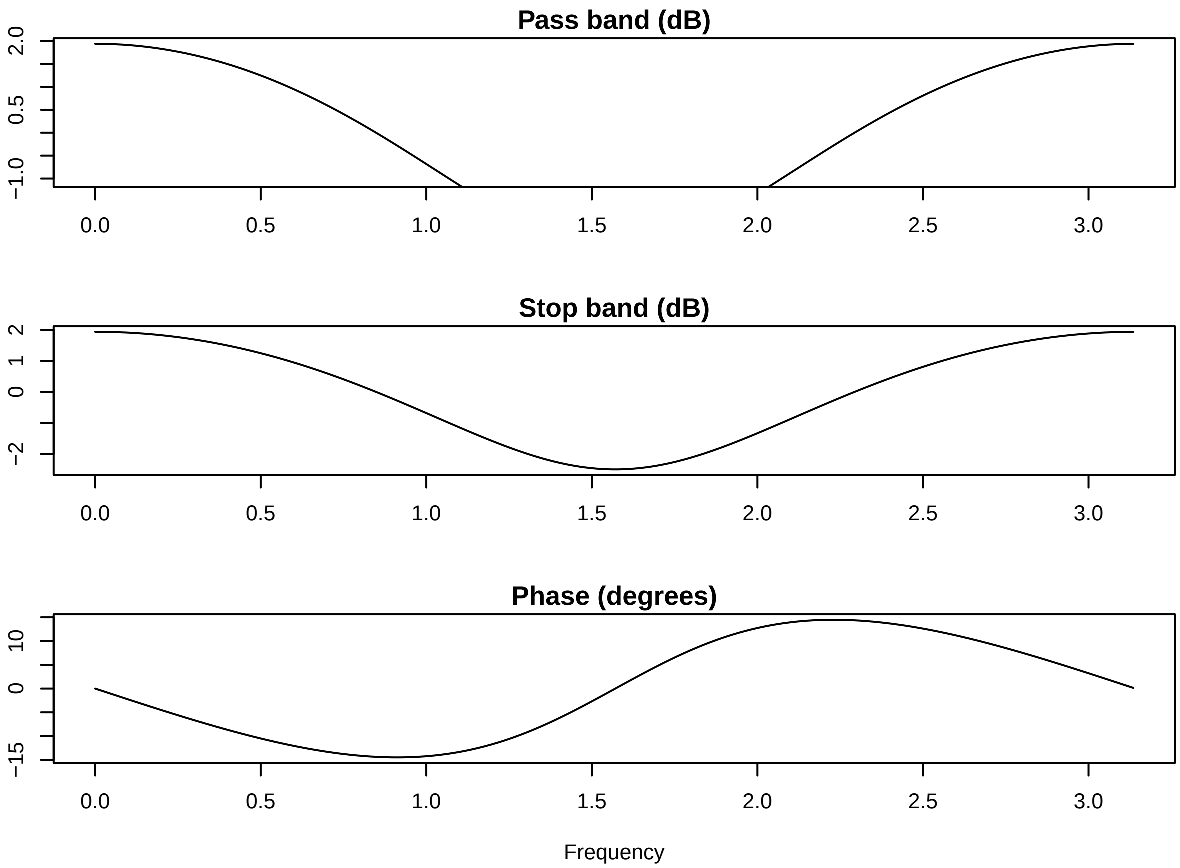
```
#Para H1:

r = 0.5
w0 = pi /2

num = c(1, -2*r*cos(w0), r^2)
den = c(1)

zplane(num,den)
```



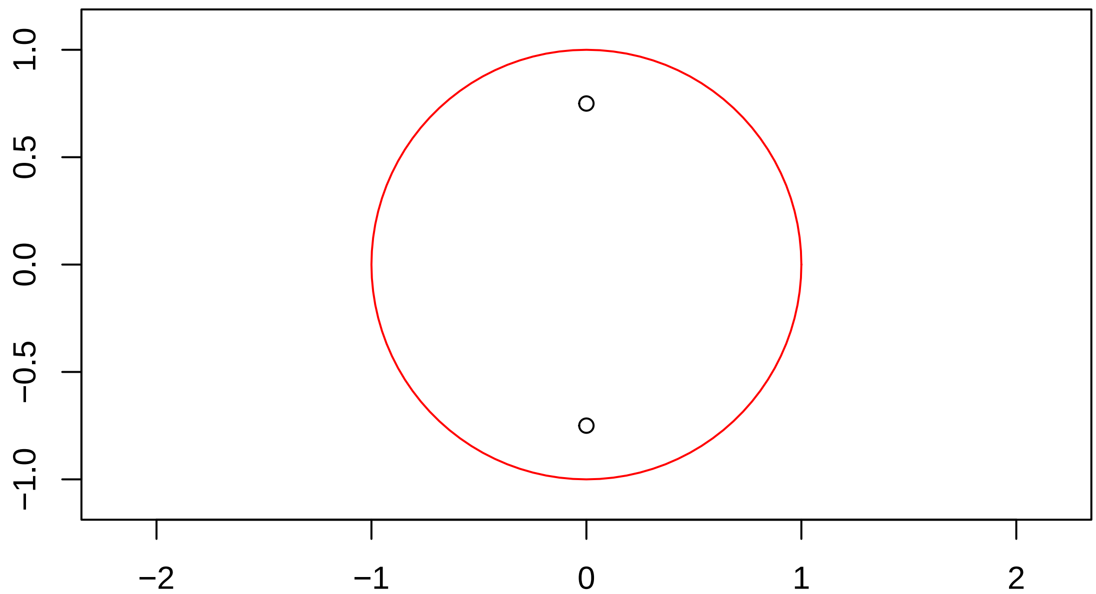

```
impz(num,den, 35)
```



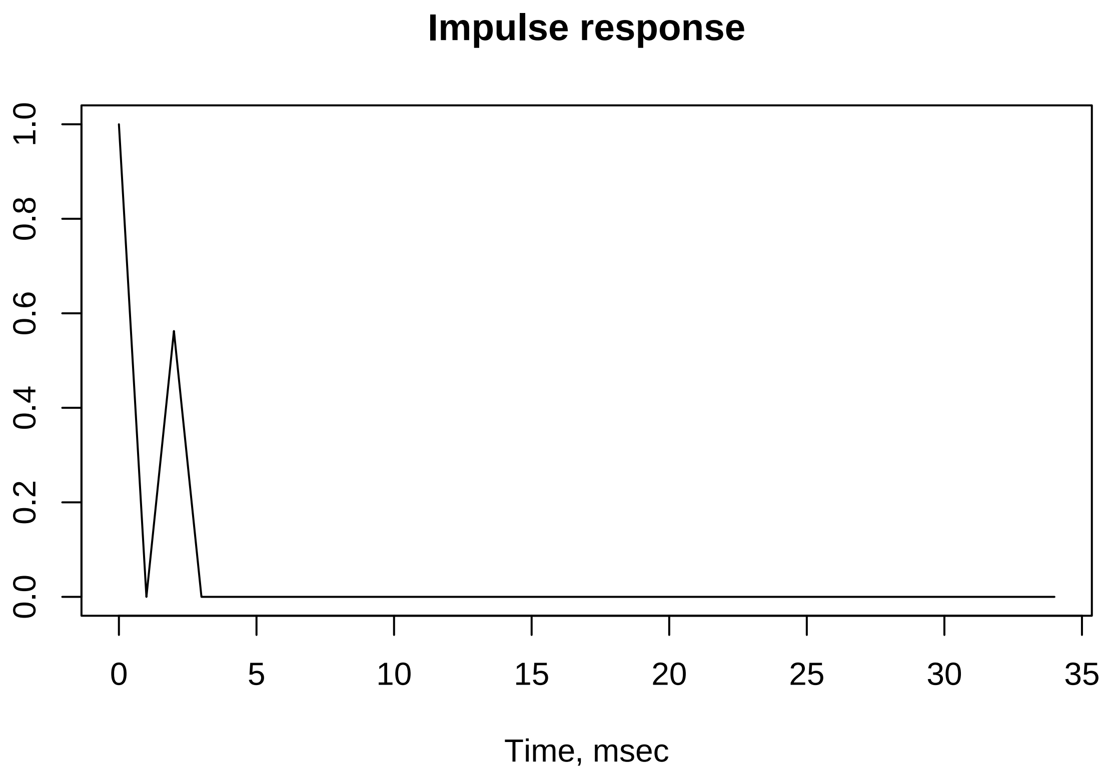

```
freqz(num,den)
```



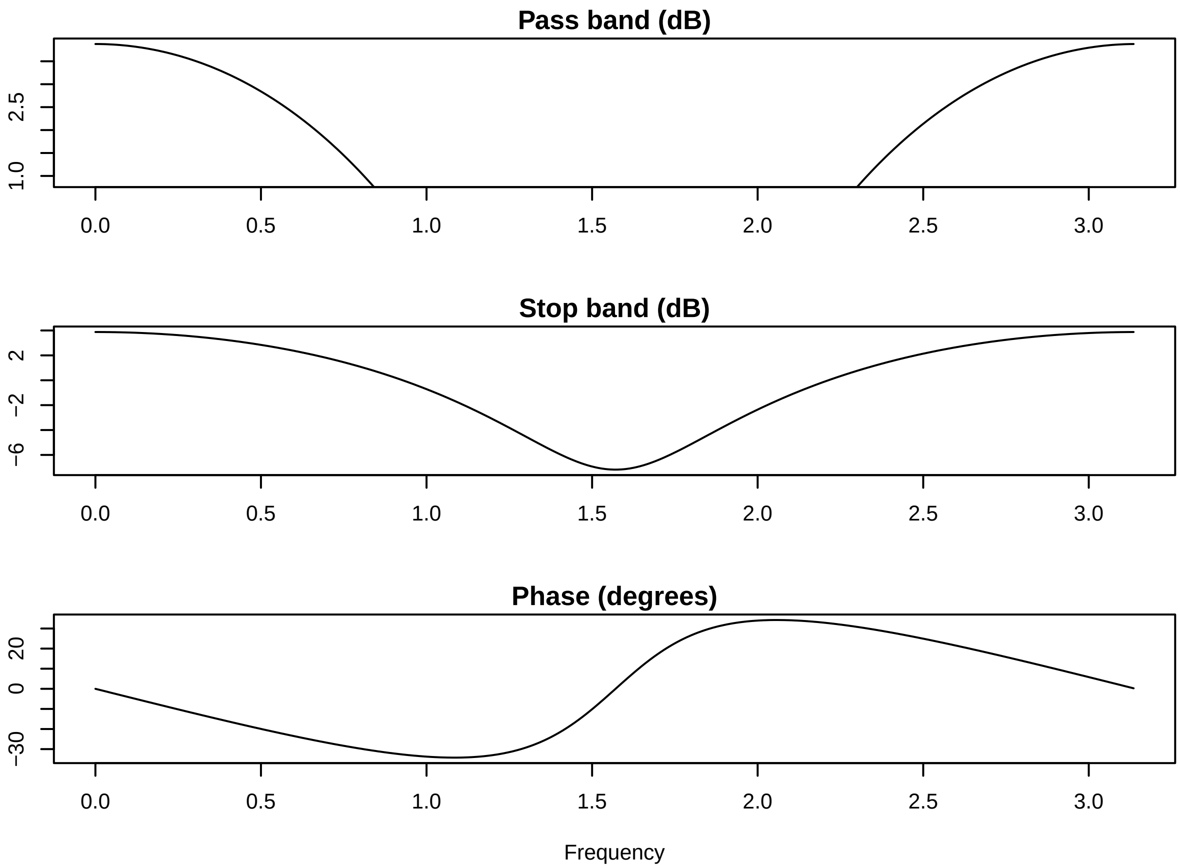

```
r = 0.75
w0 = pi/2

num = c(1, -2*r*cos(w0), r^2)
den = c(1)

zplane(num,den)
```



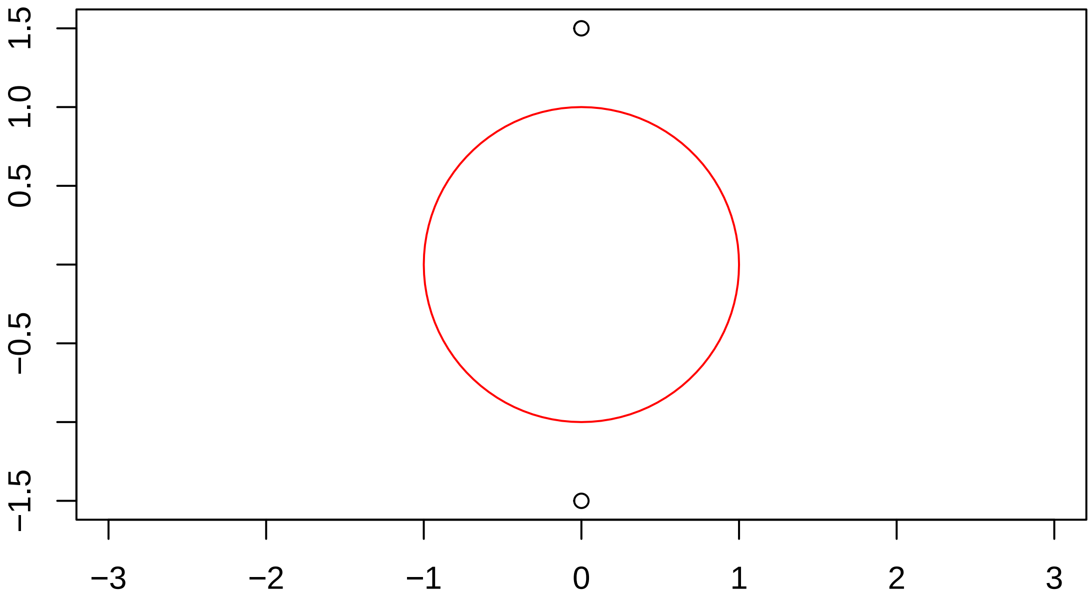

```
impz(num,den, 35)
```



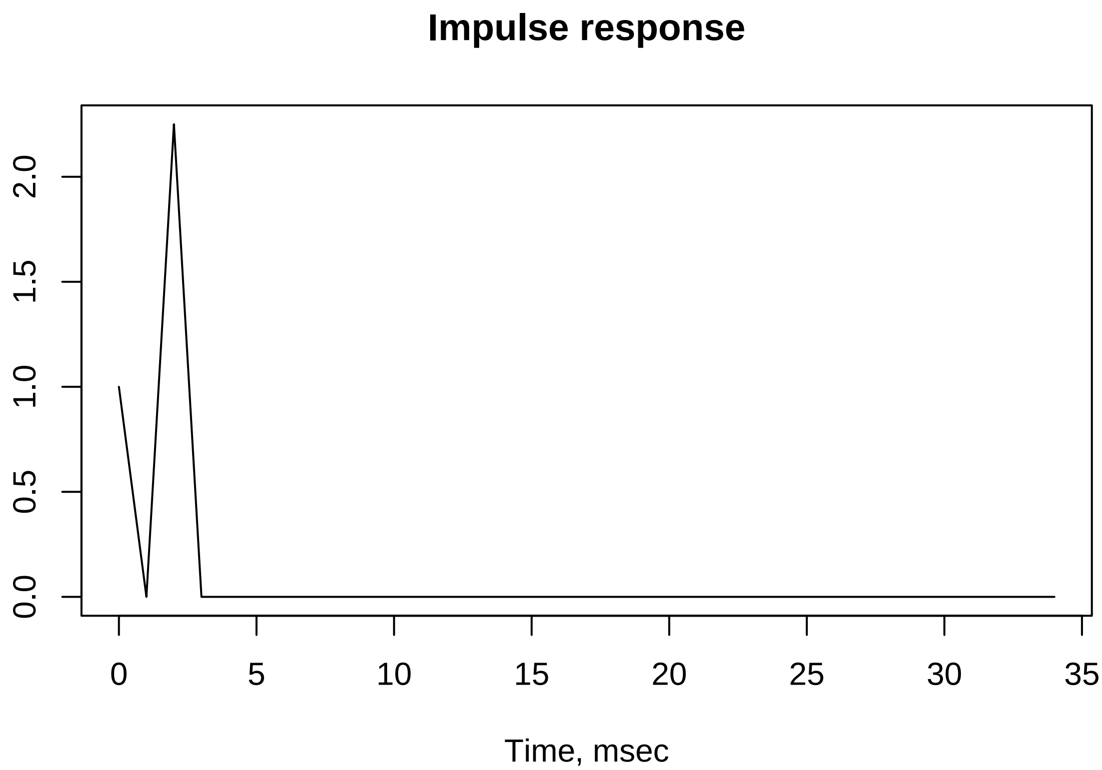

```
freqz(num,den)
```



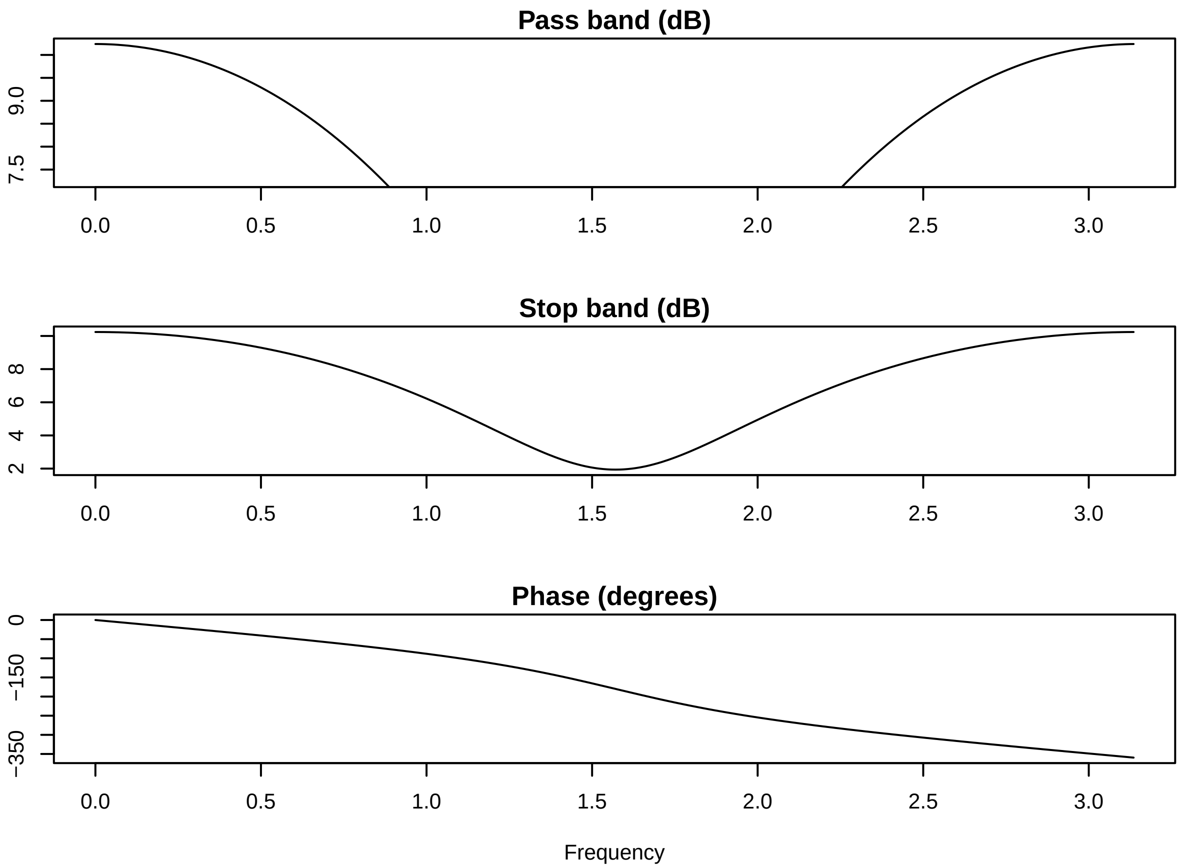

```
r = 1.5
w0 = pi /2

num = c(1, -2*r*cos(w0), r^2)
den = c(1)

zplane(num,den)
```



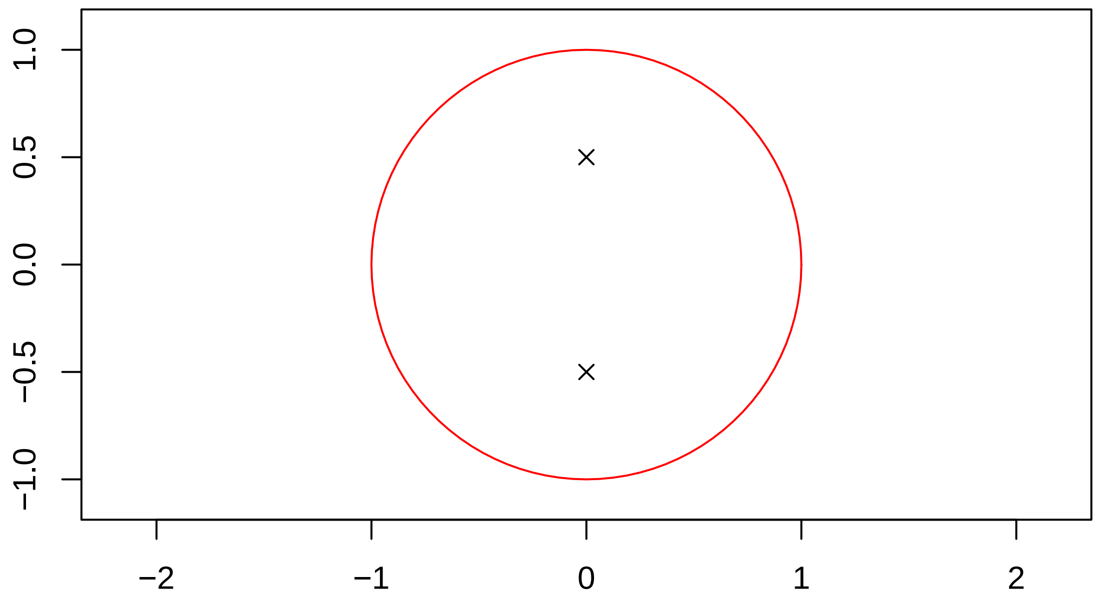

```
impz(num,den, 35)
```



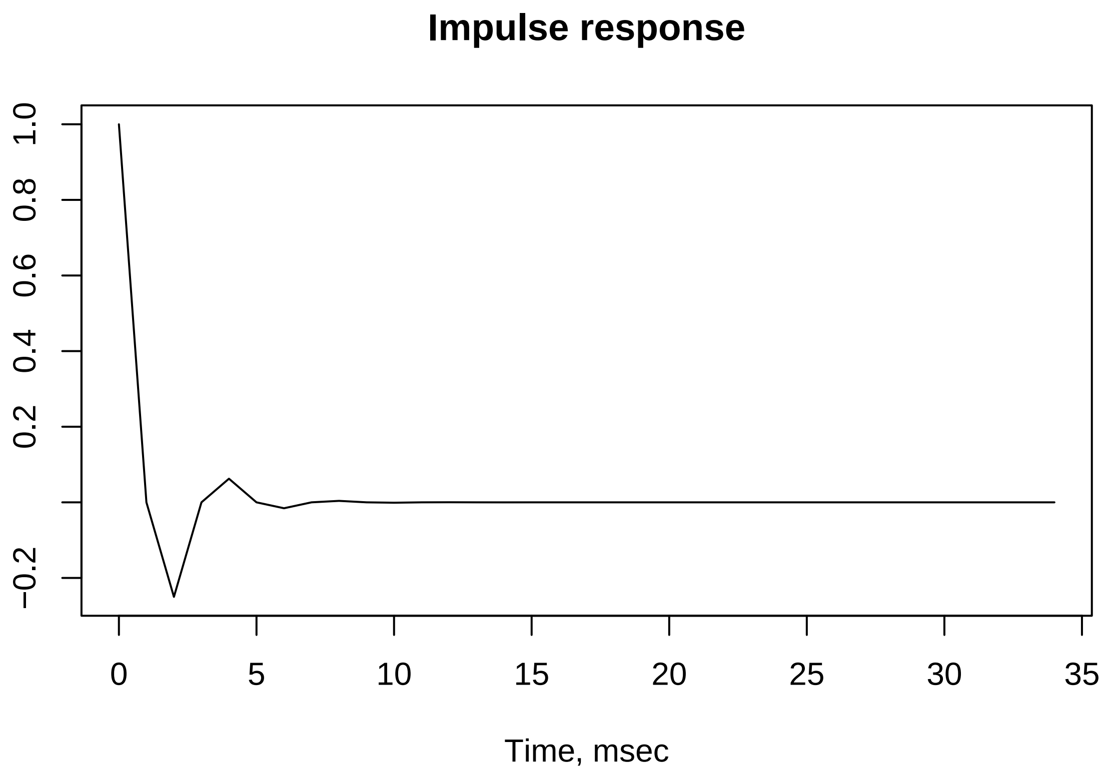

```
freqz(num,den)
```



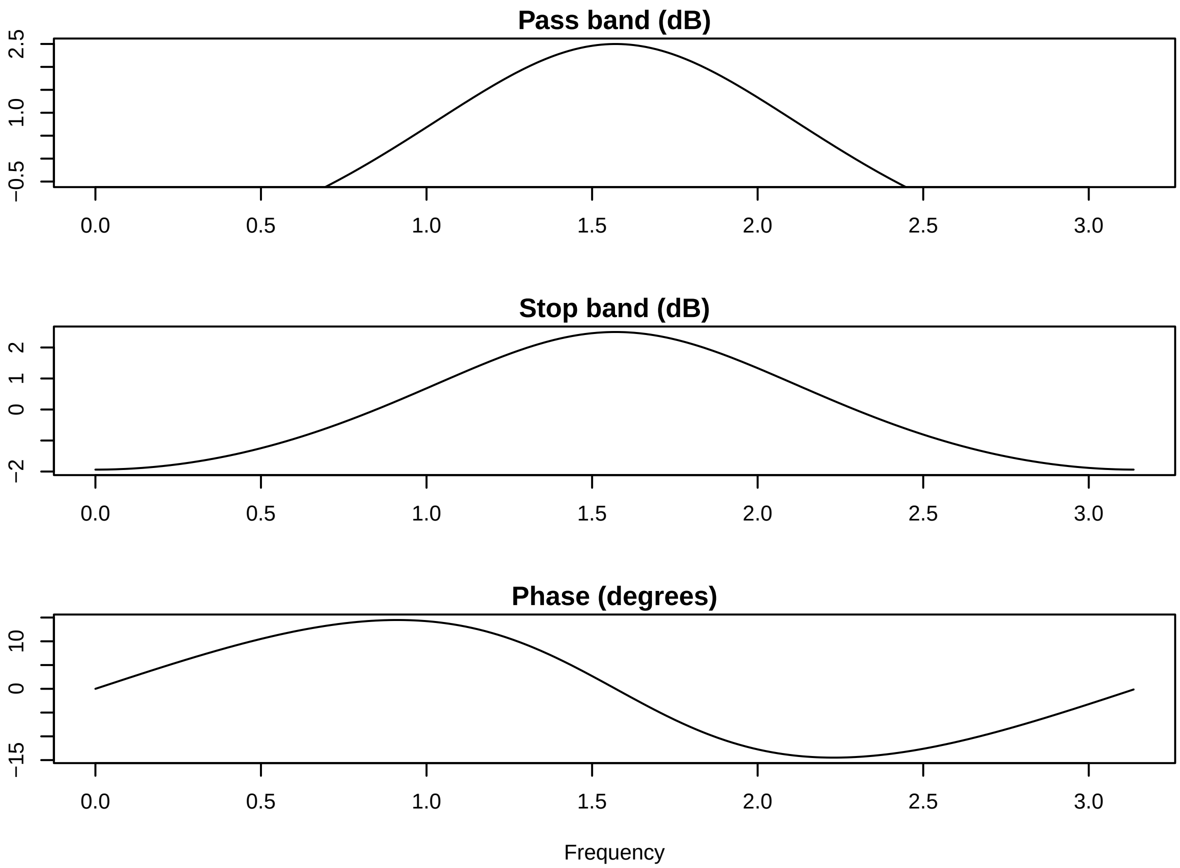

```
#Para H2:

r = 0.5
w0 = pi /2

num = c(1)
den = c(1, -2*r*cos(w0), r^2)

zplane(num,den)
```



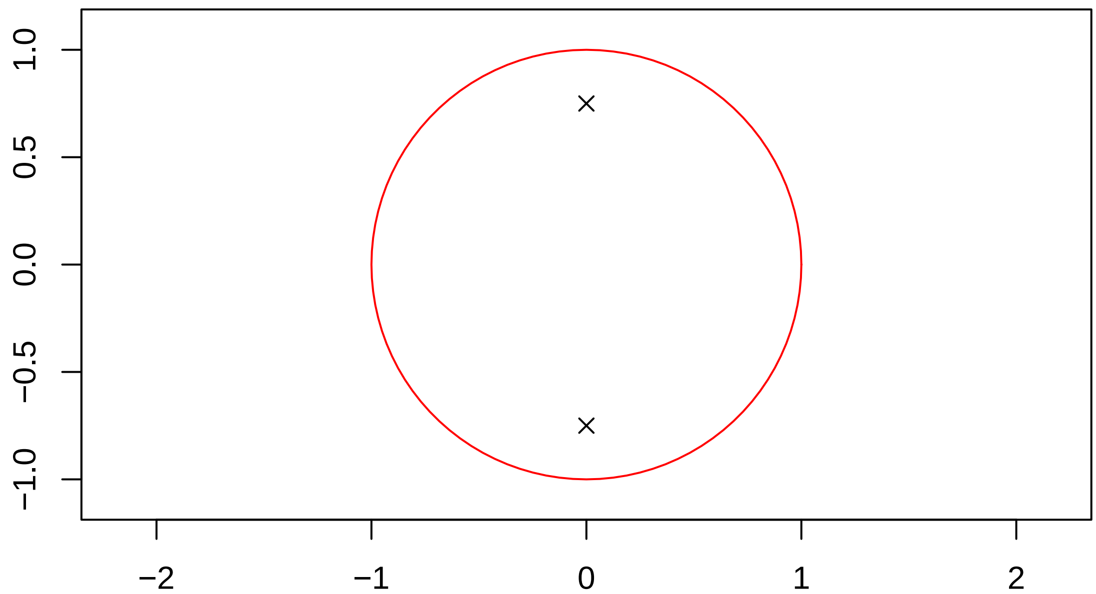

```
impz(num,den, 35)
```



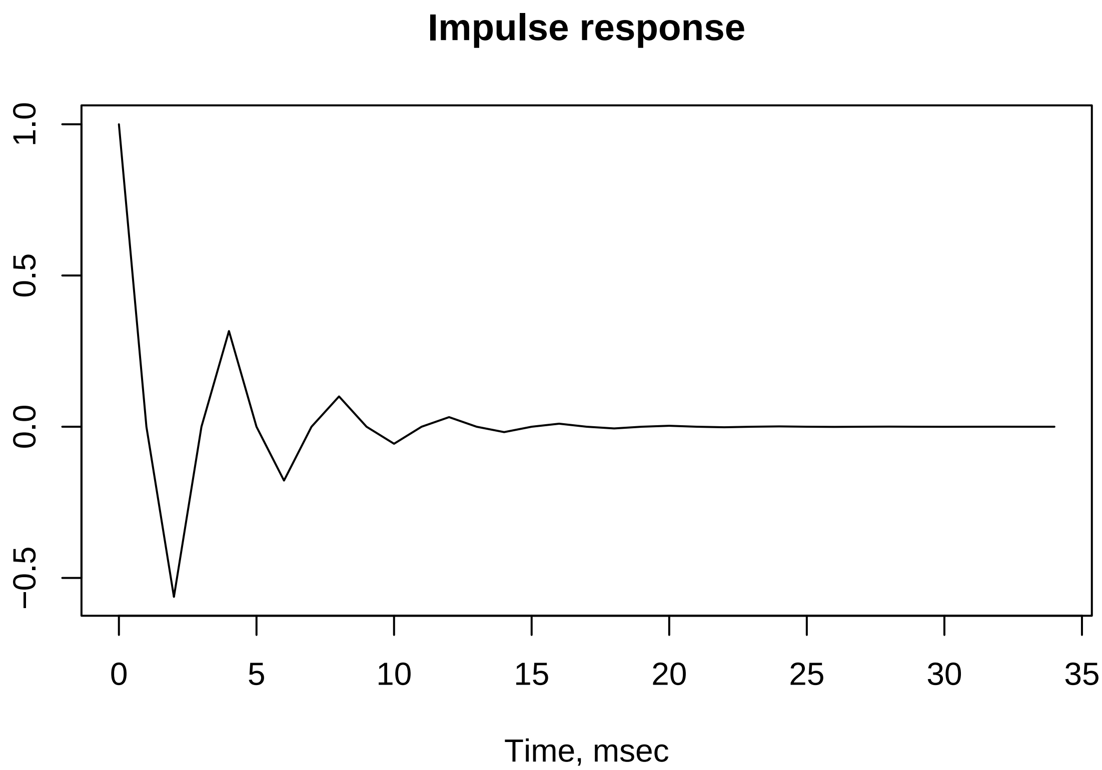

```
freqz(num,den)
```



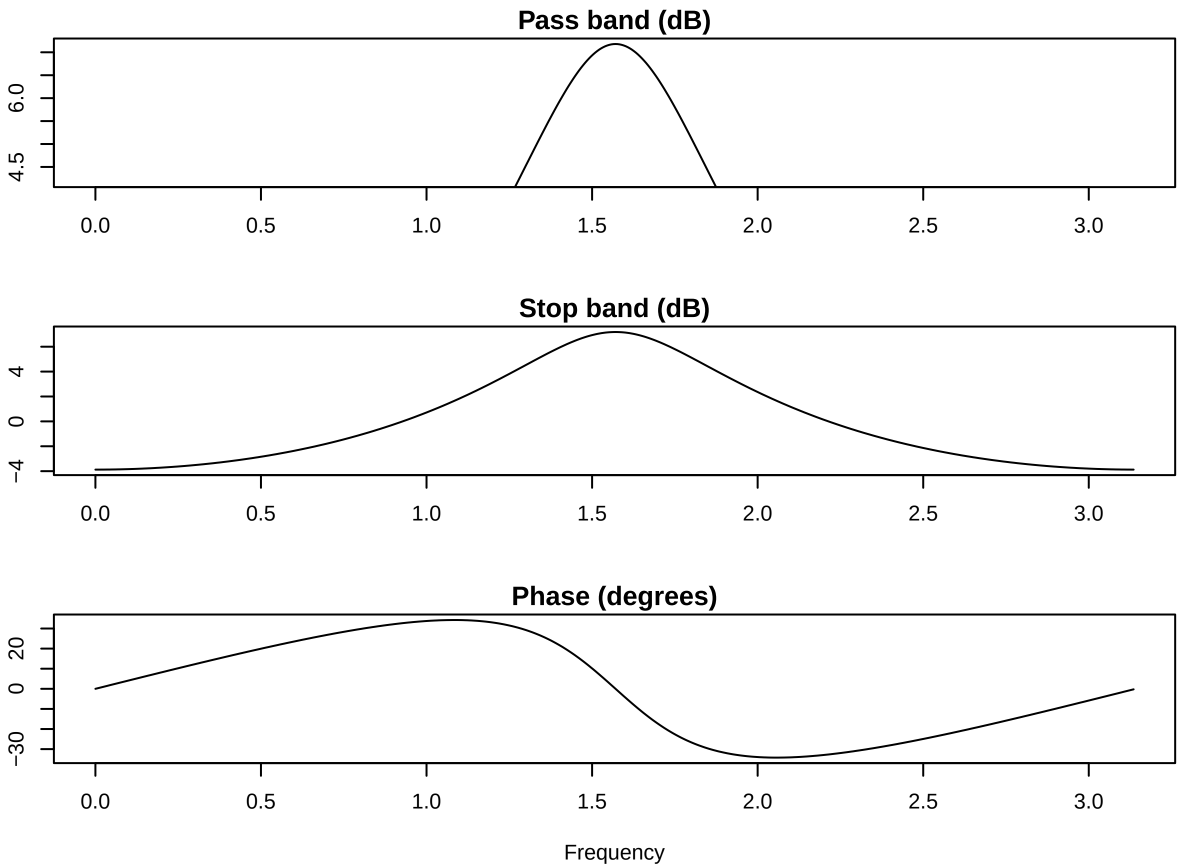

```r
r = 0.75
w0 = pi/2

num = c(1)
den = c(1, -2*r*cos(w0), r^2)

zplane(num,den)
```



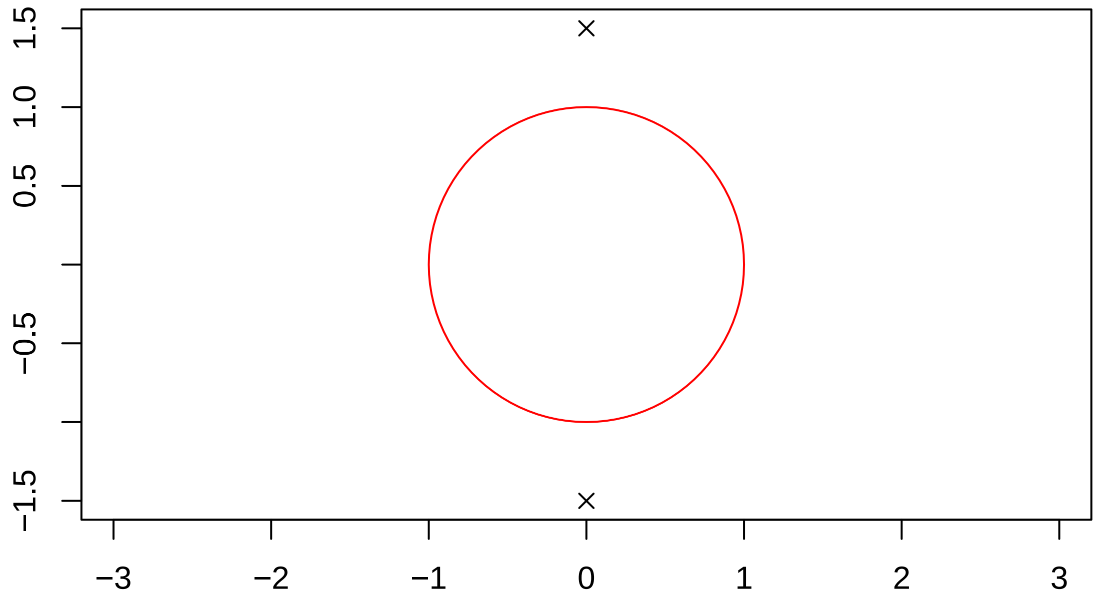

```
impz(num,den, 35)
```



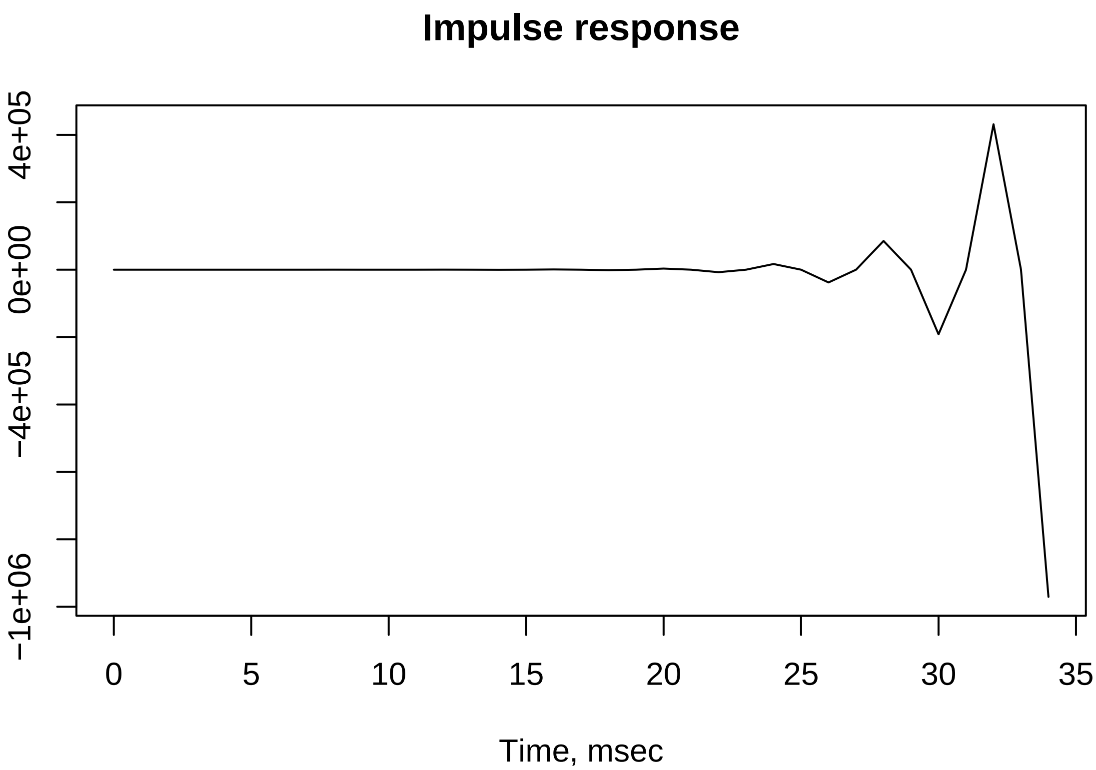

```
freqz(num,den)
```



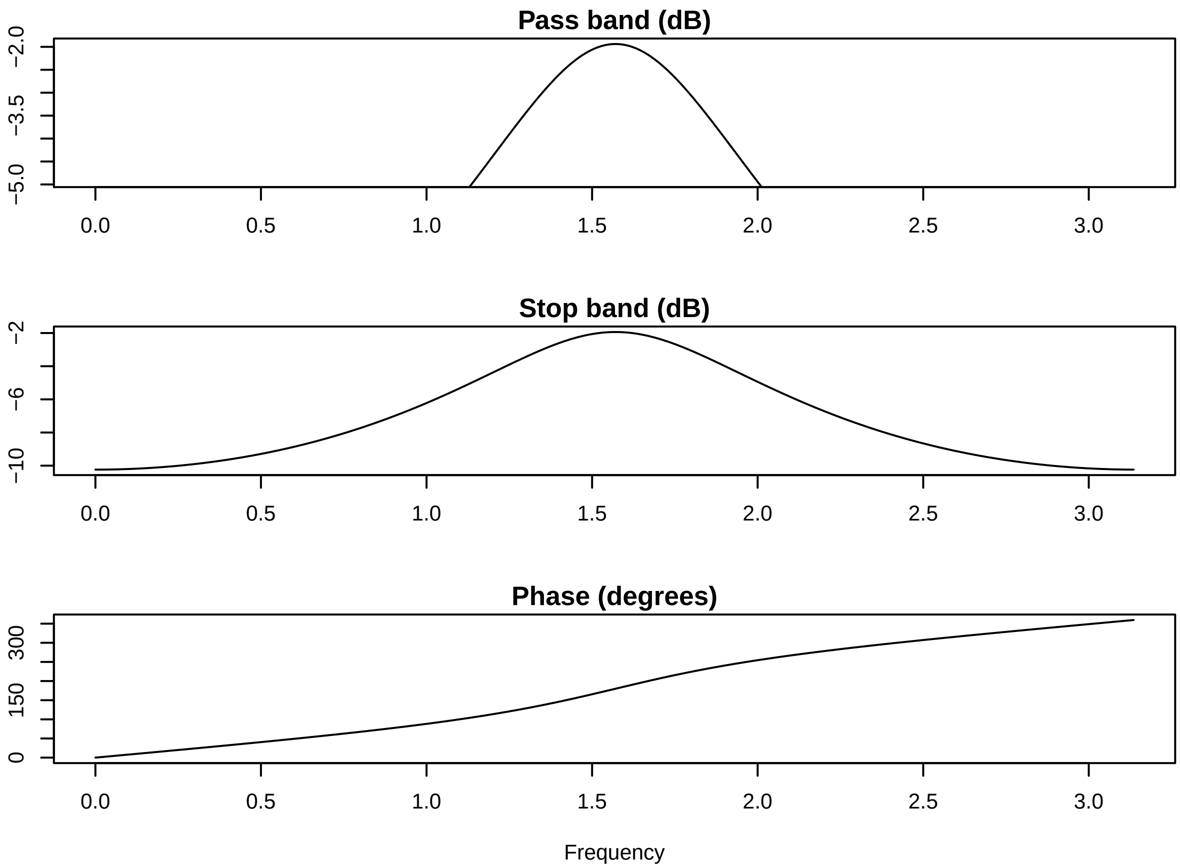

```
r = 1.5
w0 = pi /2

num = c(1)
den = c(1, -2*r*cos(w0), r^2)

zplane(num,den)
```




```
impz(num,den, 35)
```




```
freqz(num,den)
```


Pass band (dB)
−2.0
−3.5
−5.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Stop band (dB)
−2
−6
−10
Phase (degrees)
300
150
0
Frequency

3. Laboratorio

3.1 Sistemas discretos

3.1.1 Filtro peine

Un sistema digital muy empleado como filtro paso-bajo es lo que se conoce como promediador. La ecuación en diferencias de este filtro es:

$$y(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(n-k)$$

Determina la respuesta impulsional y la respuesta en frecuencia del sistema para diferentes N (2,4 y 9). Calcula los polos y ceros de $H(z)$, ¿qué tienen de especial?¿Qué ocurre en los ceros de $H(z)$?

La ecuación en diferencias desarrollada será:

$$y(n) = \frac{1}{N}(x(n) + x(n-1) + x(n-2) + ... + x(n-N-1))$$

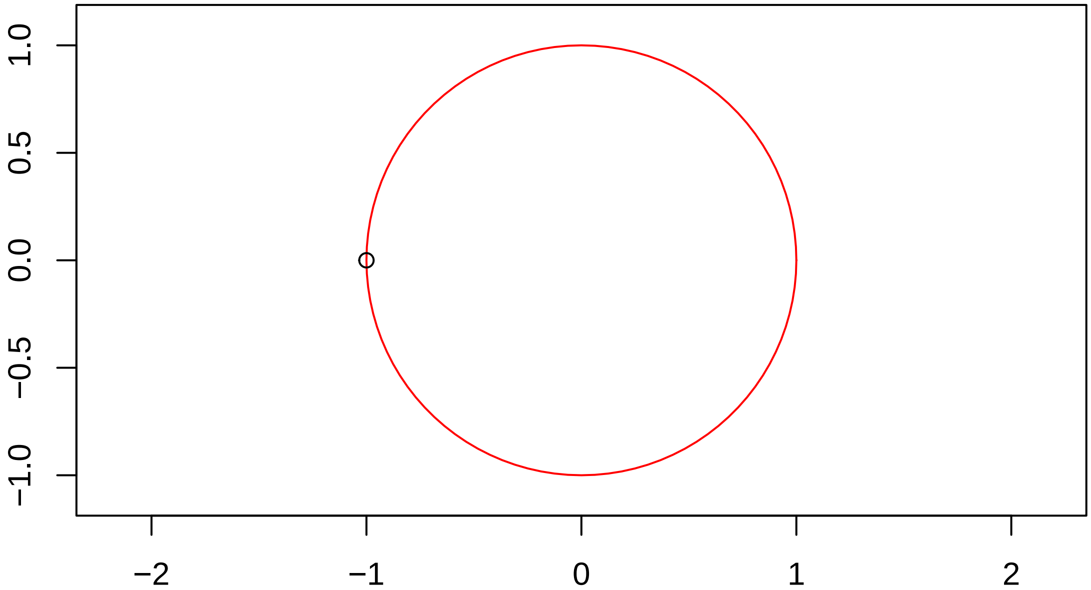
Y la funcion de tranferencia:

$$H(z) = \frac{1}{N}(1 + z^{-1} + z^{-2} + ... + z^{-N-1})$$

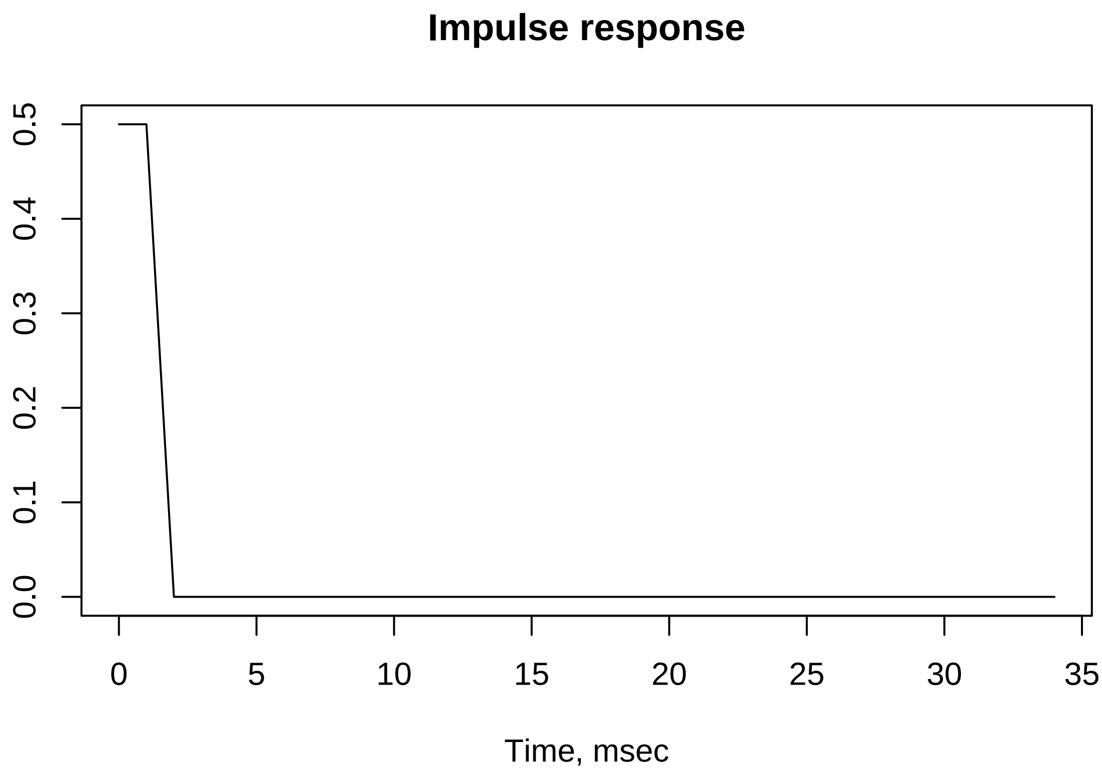
Para N=2

Como tenemos dos retrasos, el vector a será c(1,1) y por tanto, al ser $\frac{1}{N}$ el vector b es el N que es este caso es 2.

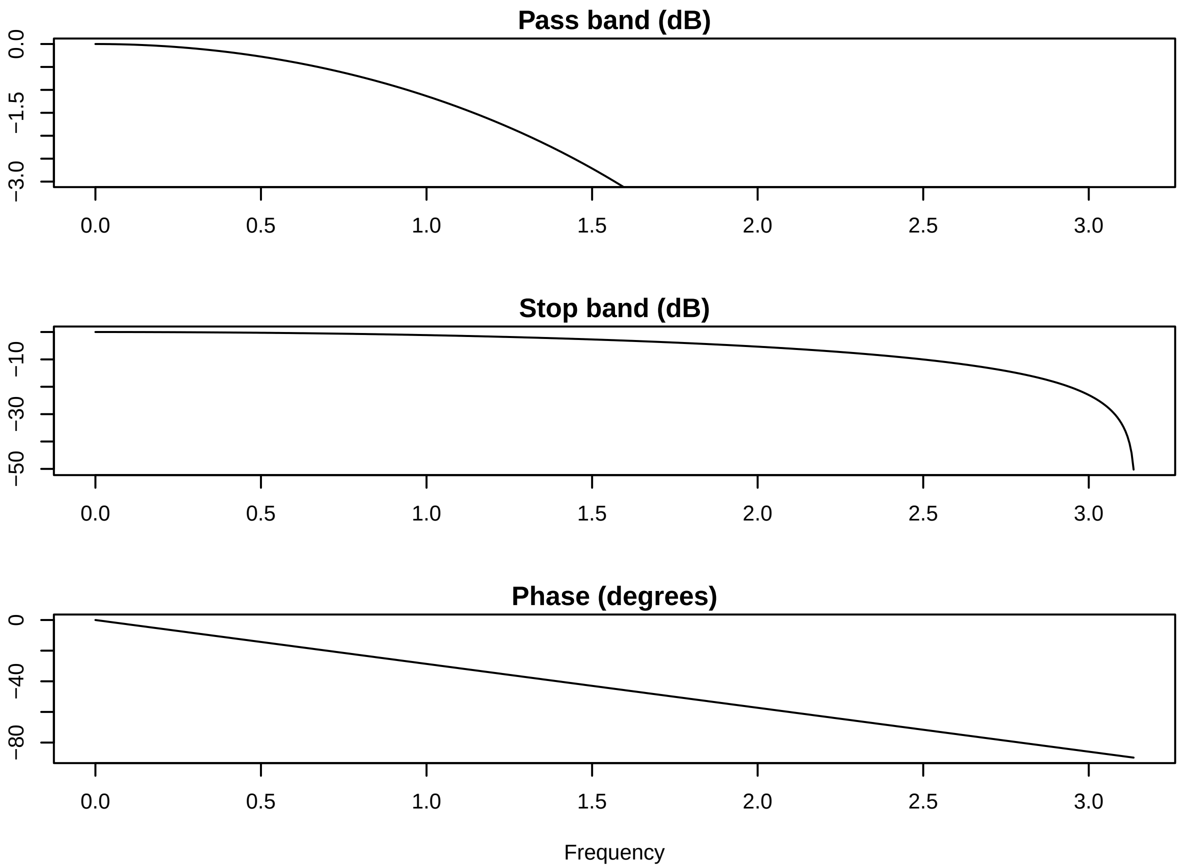
```
a = c(1,1)
b = c(2)

zplane(a,b)
```




```
impz(a,b, 35)
```

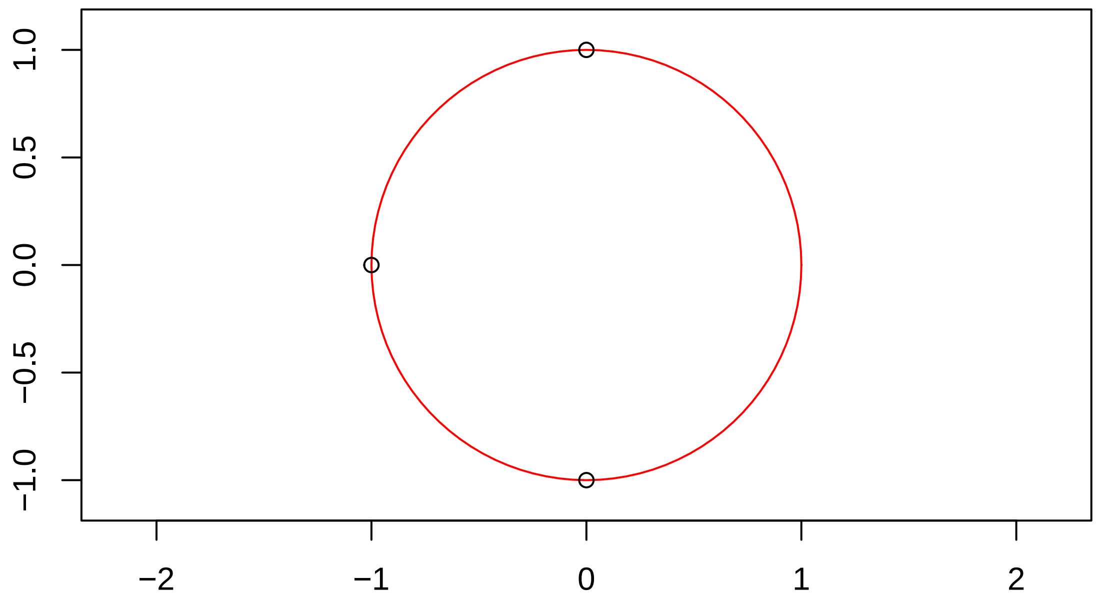



```
freqz(a,b)
```

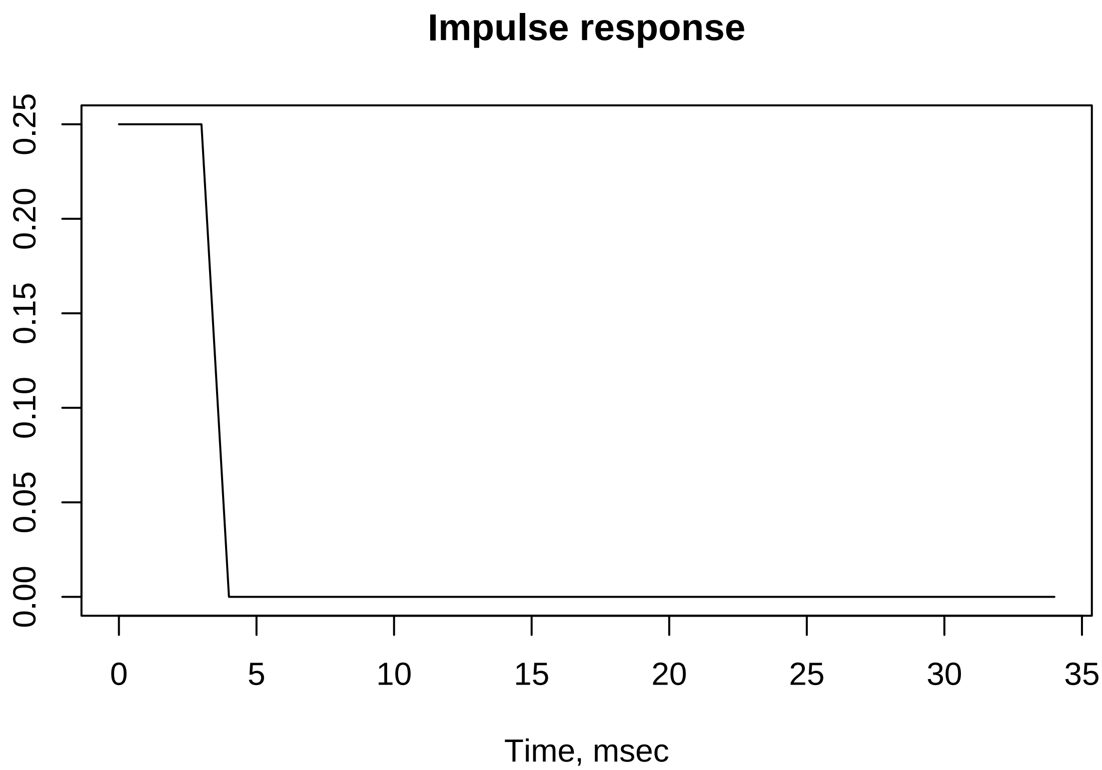


Para N=4 Como tenemos 4 retrasos, el vector a sera c(1,1,1,1) y por tanto, al ser $\frac{1}{N}$ el vector b es el N que es este caso es 4.

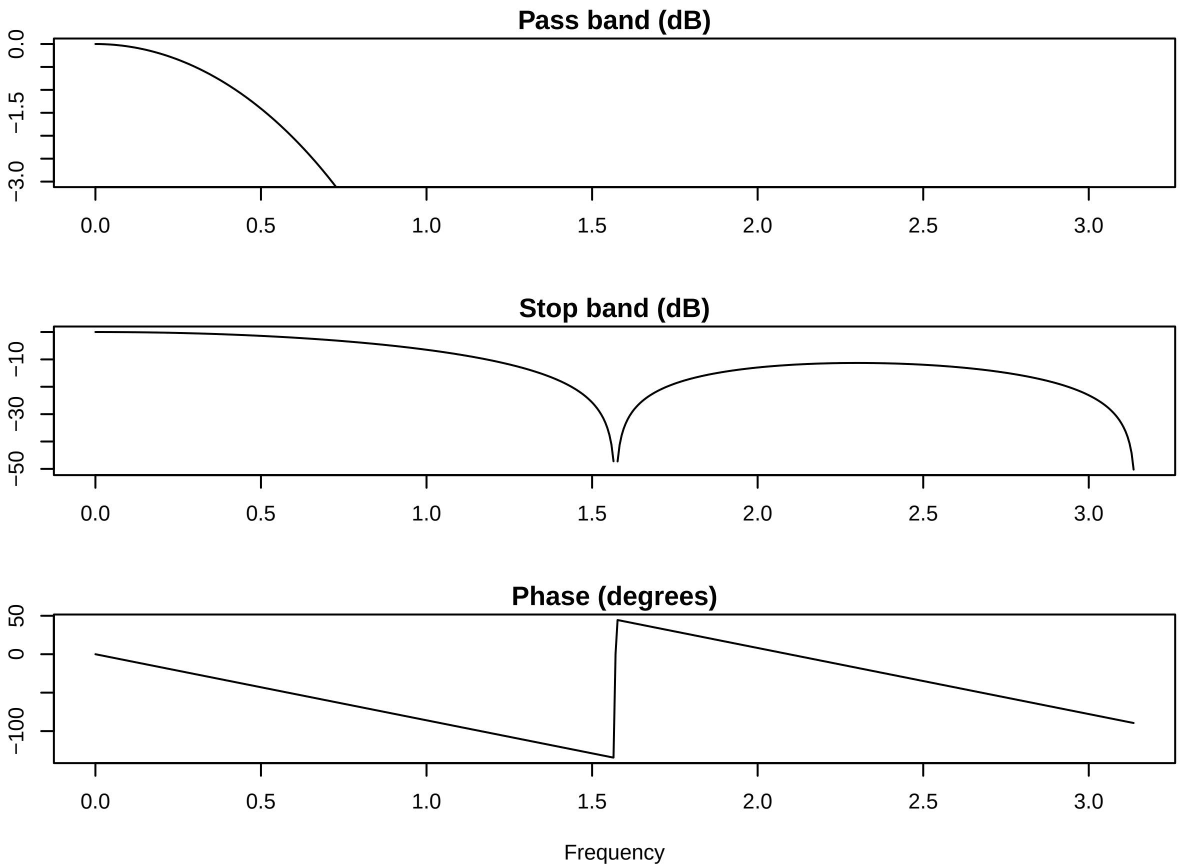
```
a = c(1,1,1,1)
b = c(4)

zplane(a,b)
```




```
impz(a,b, 35)
```


Impulse response
0.00
0.05
0.10
0.15
0.20
0.25
0
5
10
15
20
25
30
35
Time, msec

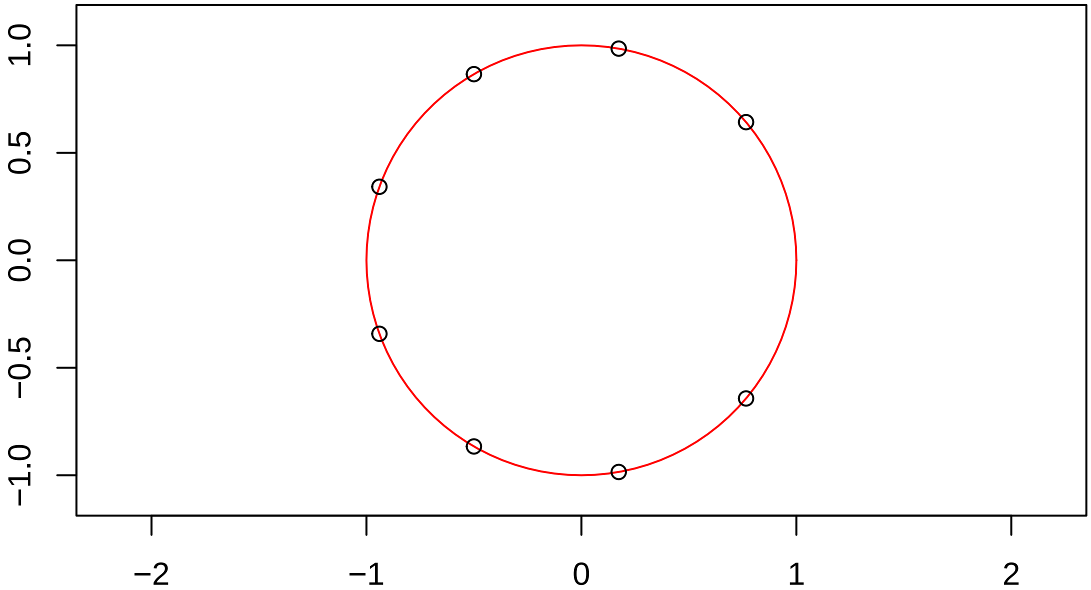
```
freqz(a,b)
```



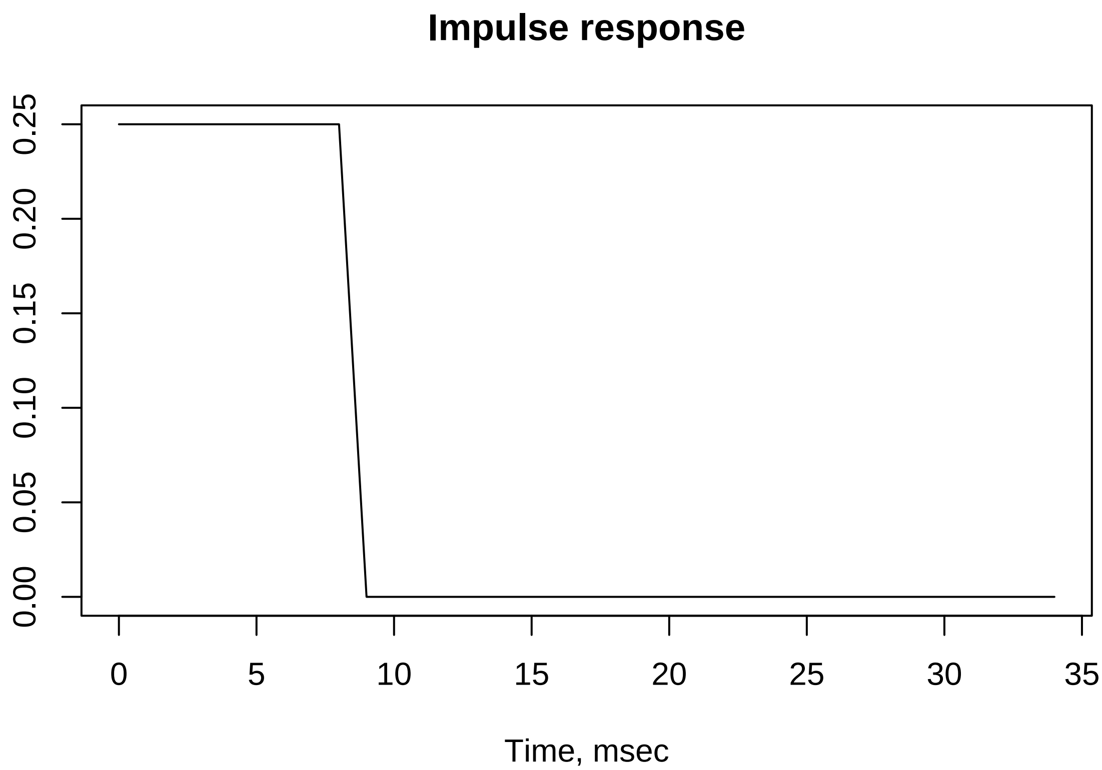
Para N=9

Como tenemos 9 retrasos, el vector a sera c(1,1,1,1,1,1,1,1,1) y por tanto, al ser $\frac{1}{N}$ el vector b es el N que es este caso es 9.

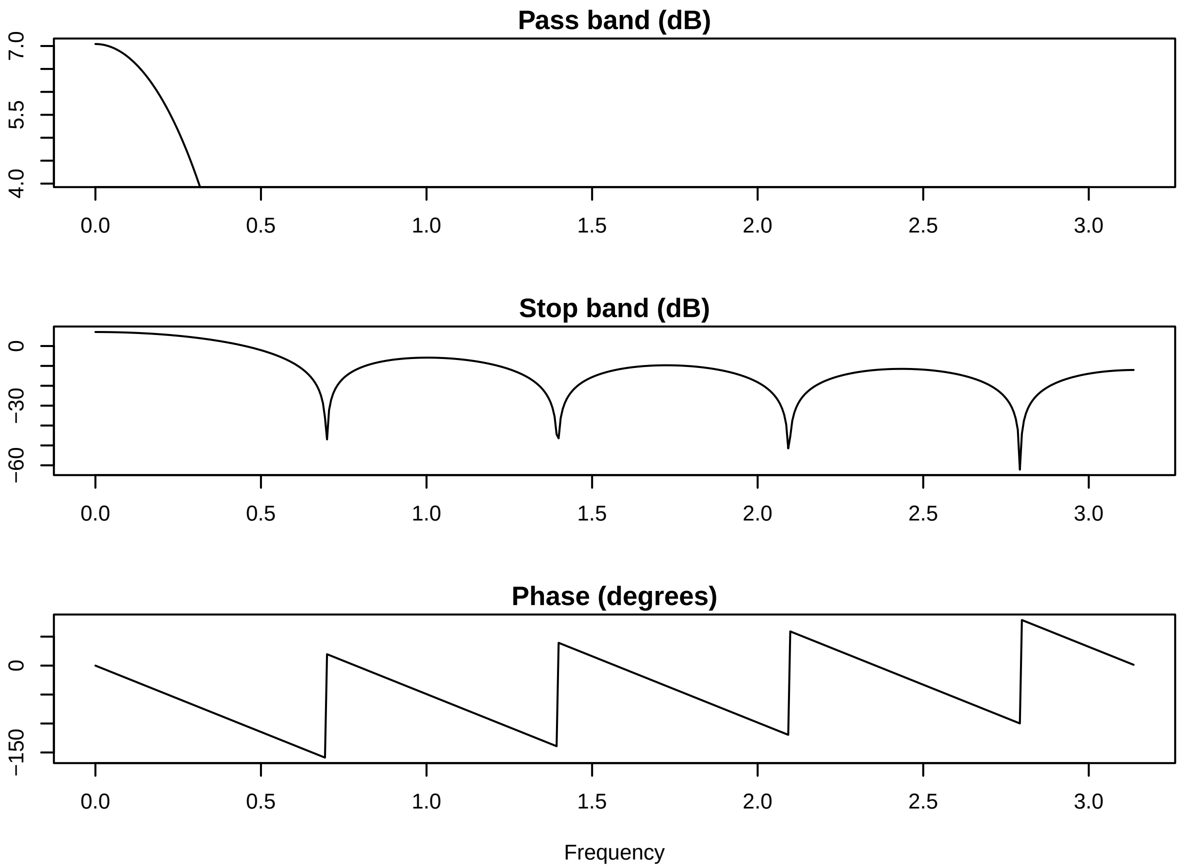
```
a = c(1,1,1,1,1,1,1,1,1)
c = c(9)

zplane(a,b)
```




```
impz(a,b, 35)
```


Impulse response
0.00
0.05
0.10
0.15
0.20
0.25
0
5
10
15
20
25
30
35
Time, msec

```
freqz(a,b)
```



CONCLUSIONES: En N= 9 obtenemos el filtro peine claramente. Con cada 0 va haciendo que el filtro sea mas paso bajo. Y eso significa que solo deja pasar las frecuencias mas cercanas a 0. Ademas, en el N=9 tenemos mas nodulos de bajada(tenemos 4 nodulos) y se atenua más en dB.

¿Que tiene de especial? Que los ceros están en el circulo unidad y los polos están en 0.

¿Que ocurre en los ceros de $H(z)$*?* Con cada 0 va haciendo que el filtro sea mas paso bajo.

3.1.2 Eliminación del ruido de red

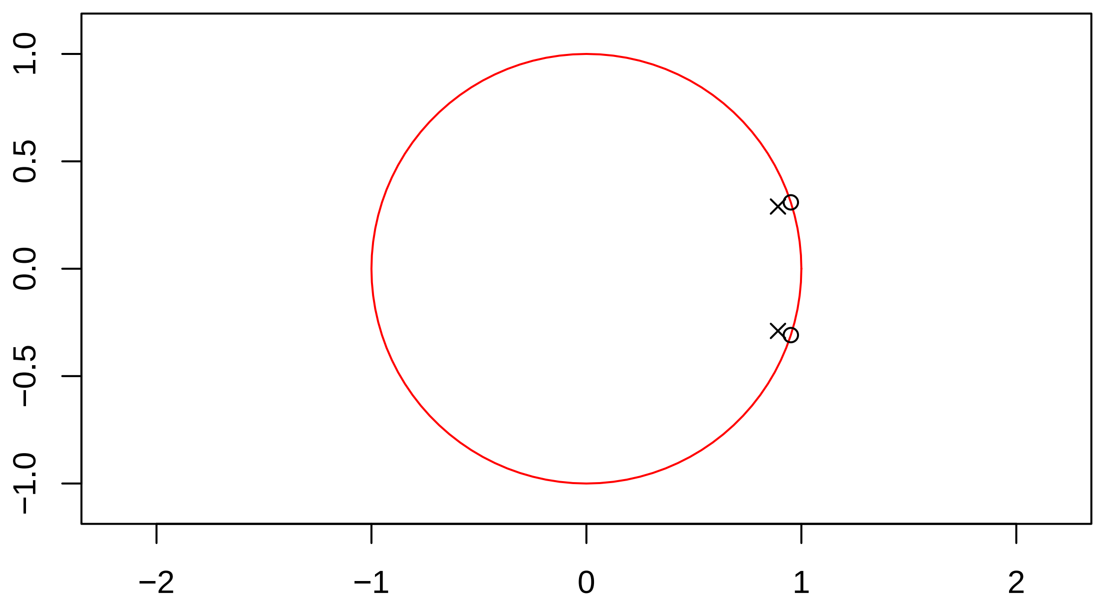
A continuación, se implementará una aplicación práctica en Ingeniería Biomédica, la eliminación del ruido de red en un electrocardiograma (ECG). Para ello utiliza el fichero `ecg.mat` que contiene un ECG contaminado por ruido de 50 Hz. Diseña un sistema digital para eliminar dicha interferencia mediante un análisis de polos/zeros de la Transformada Z (el ECG se ha adquirido con una frecuencia de muestreo de ECG de 1000 Hz). Implementa diferentes alternativas para eliminar este ruido.

En este ECG aparece otra interferencia importante que es una variación de la línea basal que aparece en bajas frecuencias. Para eliminarla, usa el sistema definido por $H(z) = \frac{1-z^{-1}}{1-\mu z^{-1}}$.

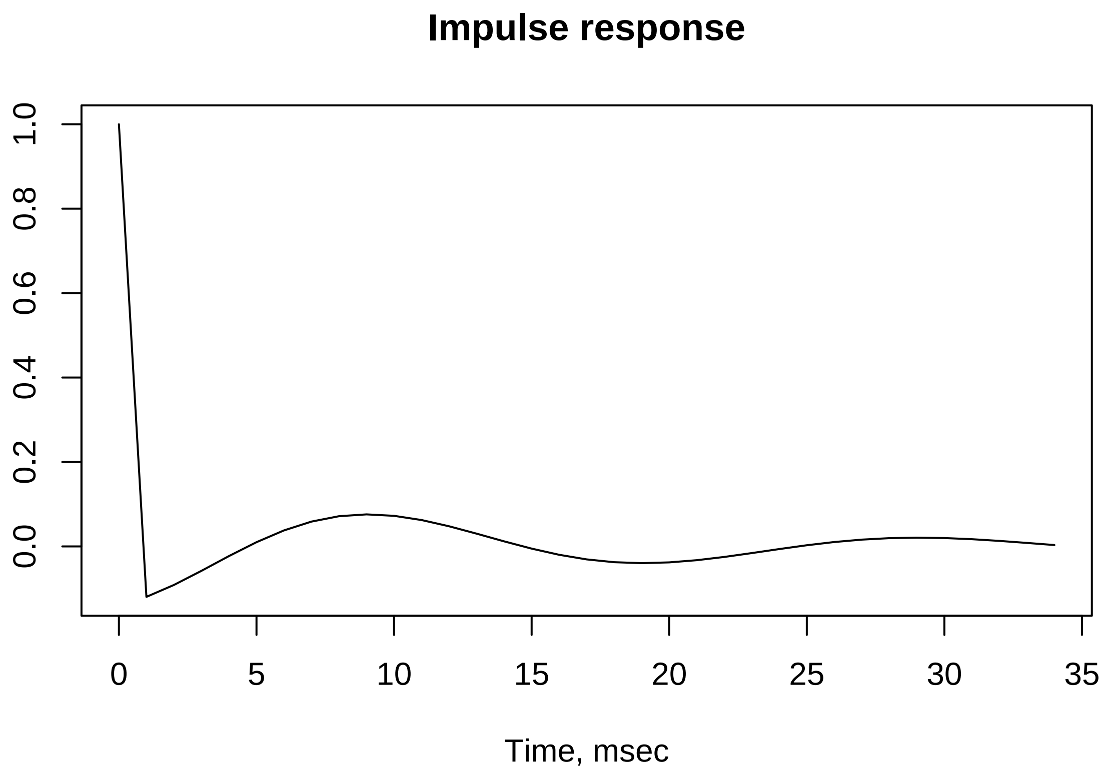
1. Compruébalo con la instrucción filter (considera $\mu = 0.9$).

```
ecg <- readMat("./ecg.mat")$ecg

ruido <- 50
muestreo <- 1000
# 2*pi*freq
```

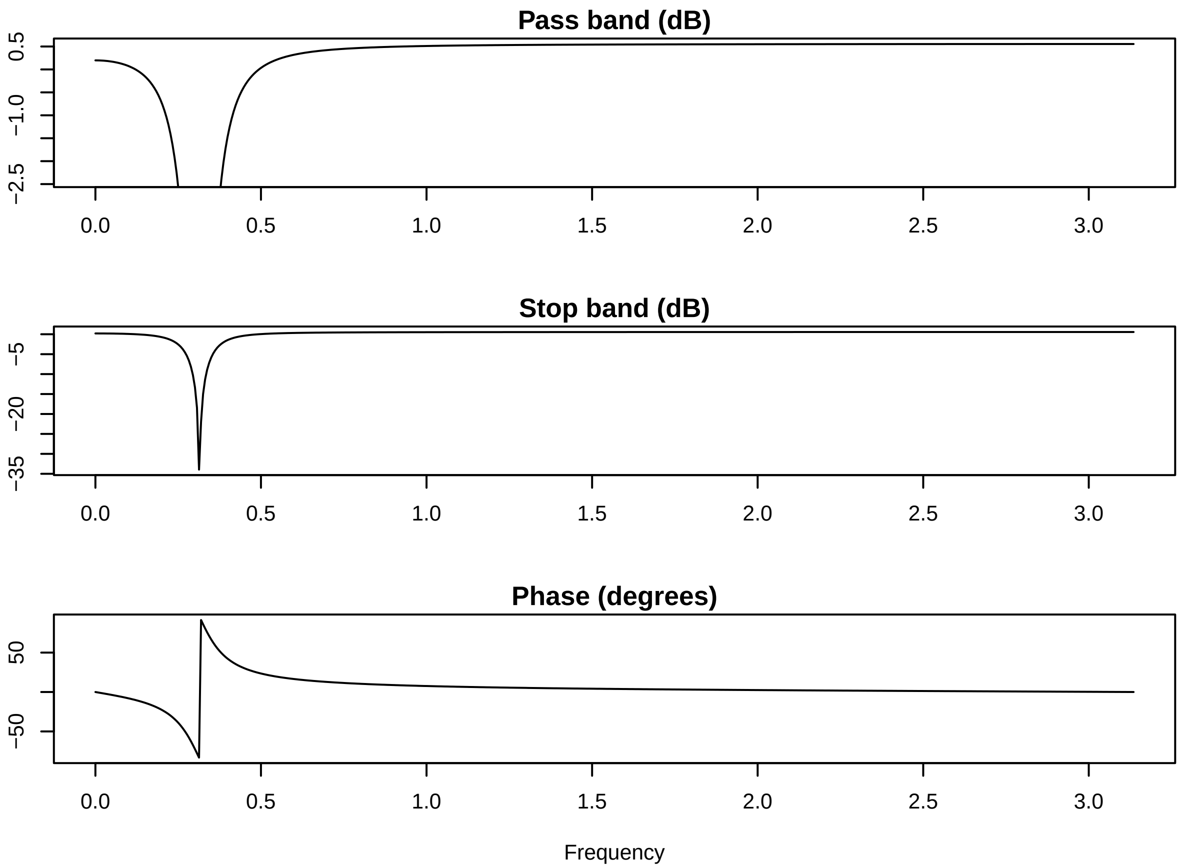
```r
w0 <- 2*pi*ruido/muestreo

#ceros
rc <- 1
# polos
rp <- 1-(2*pi)/100

x <- c(1,0,0,0,0,0,0)

num <- c(1, -2*rc*cos(w0), rc^2)
den <- c(1, -2*rp*cos(w0), rp^2)

zplane(num, den)
```



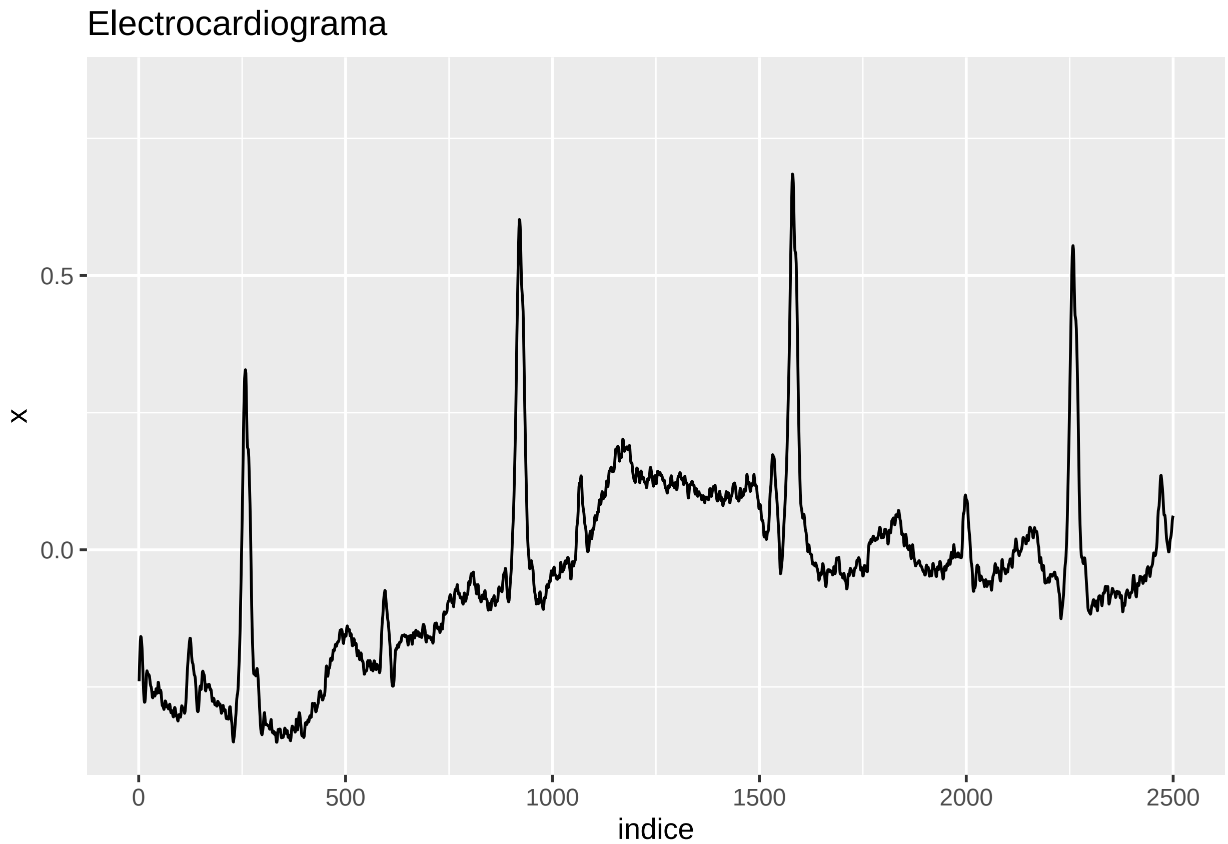

```r
impz(num, den, 35)
```



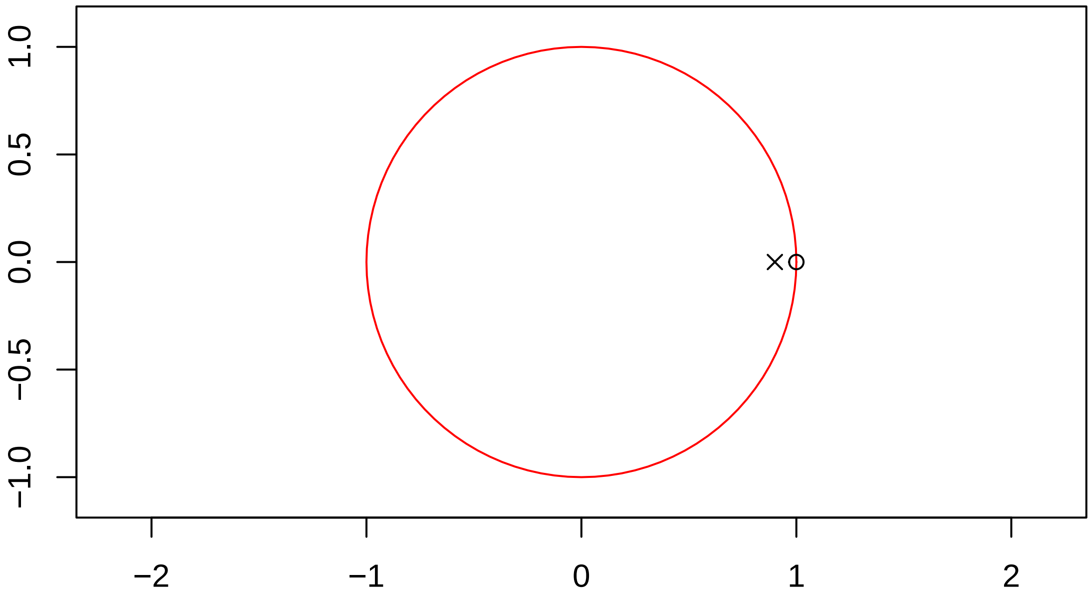

```
freqz(num, den)
```



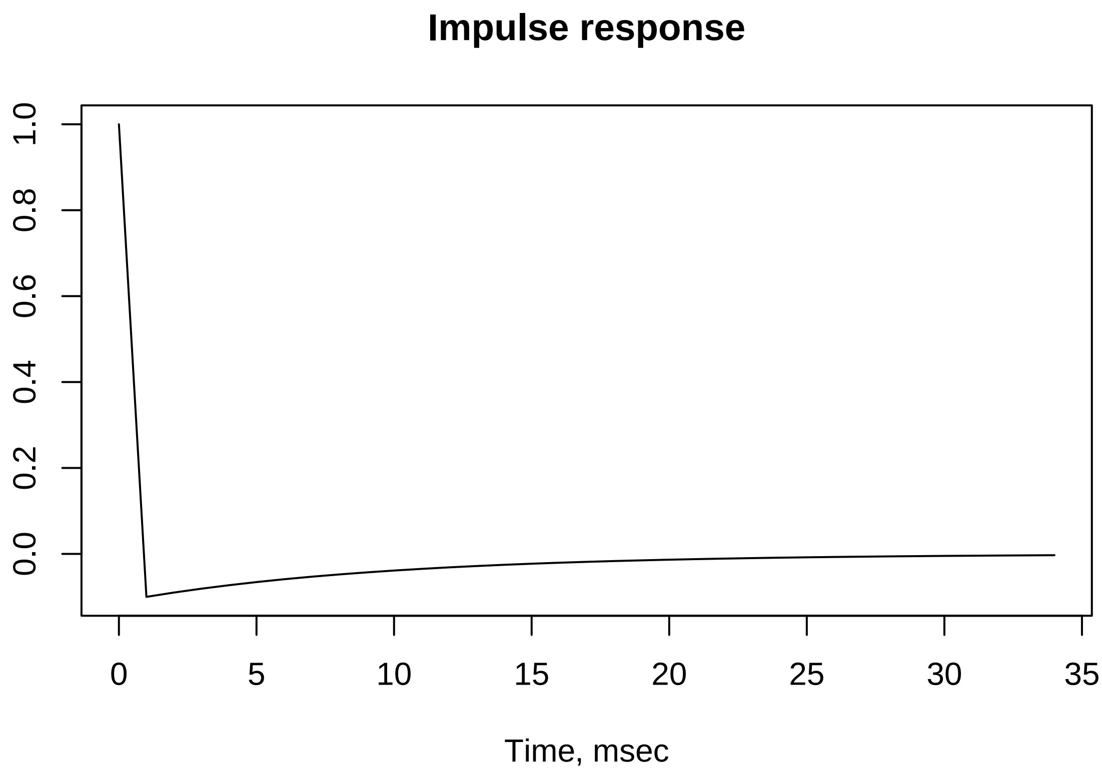

```
p1 <- filter(num, den, ecg)

# Representamos el ECG tras aplicar el primer filtro


y1_2 <- as.data.frame(p1) # pasamos a dataframe la variable que queremos representar
indice <- c(1:nrow(y1_2)) #creamos un vector con el número de las filas que hay en
# nuestro df
y1_2 <- cbind(indice, y1_2) # unimos el vector del número de filas con la variable
# que queremos representar
y1_2 <- as.data.frame(y1_2) # lo convertimos todo a dataframe

ggplot(y1_2) + geom_line(aes(x=indice, y=x)) + xlim(0,2500) + ggtitle("Electrocardiograma")
```



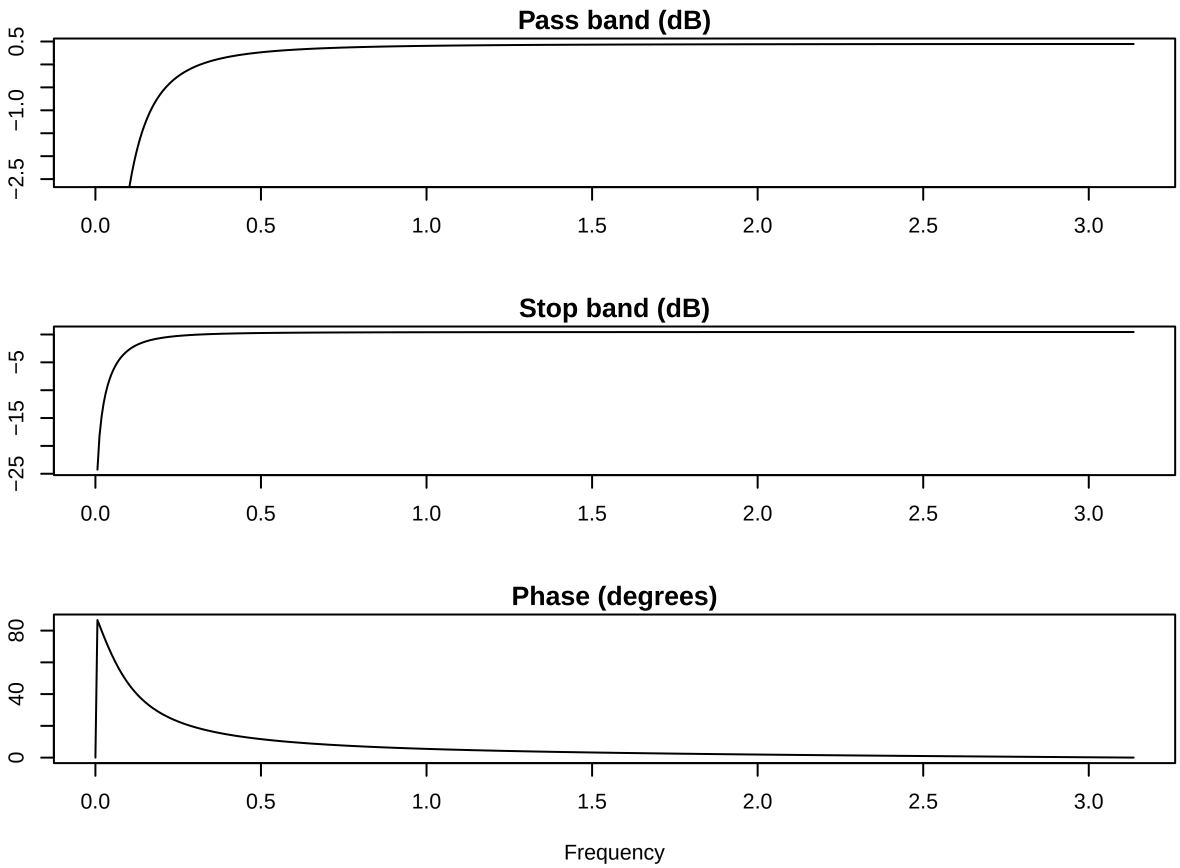

```
# Aplicamos el segundo filtro para distintos valores de mu y representamos la
# señal con los dos filtros

mu <- 0.9

num1 <- c(1,-1)
den1 <- c(1, -mu)

salida <- filter(num1, den1, p1)
zplane(num1, den1)
```



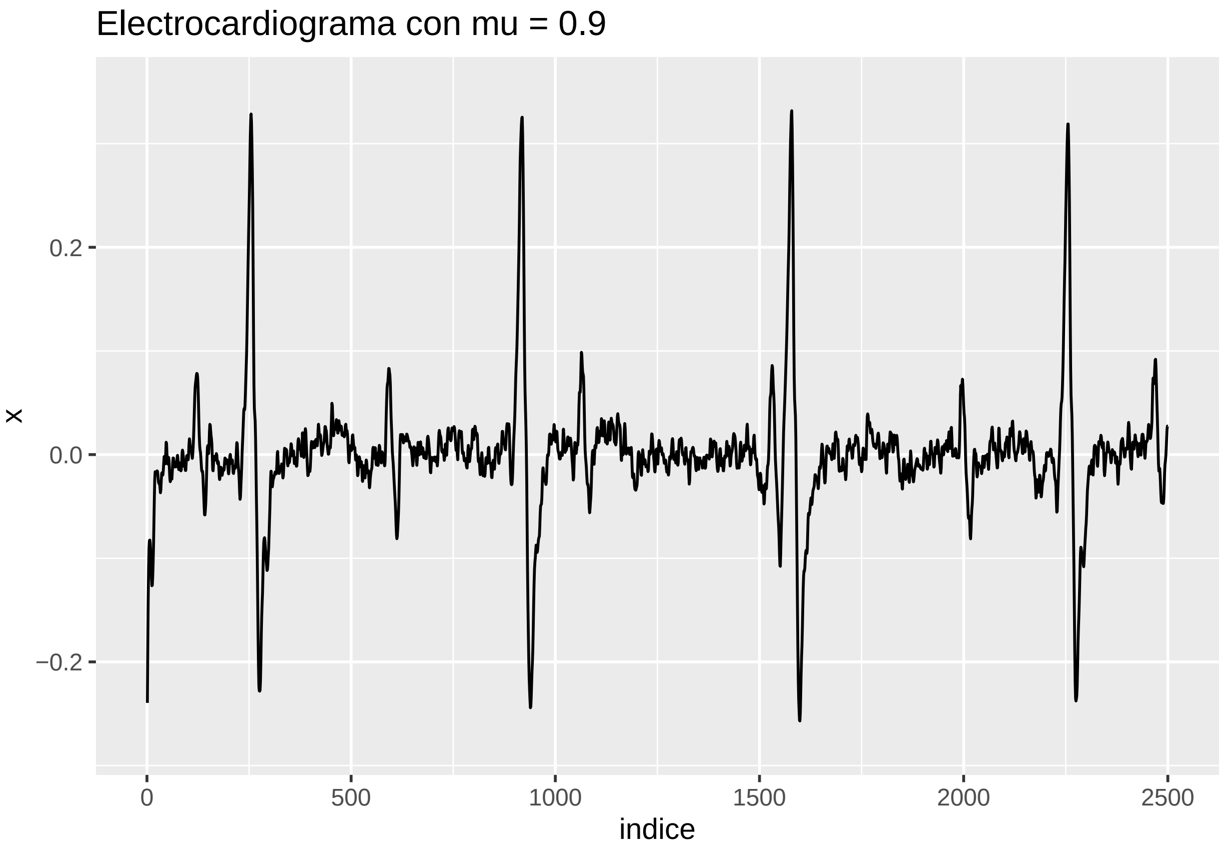

```
impz(num1, den1, 35)
```



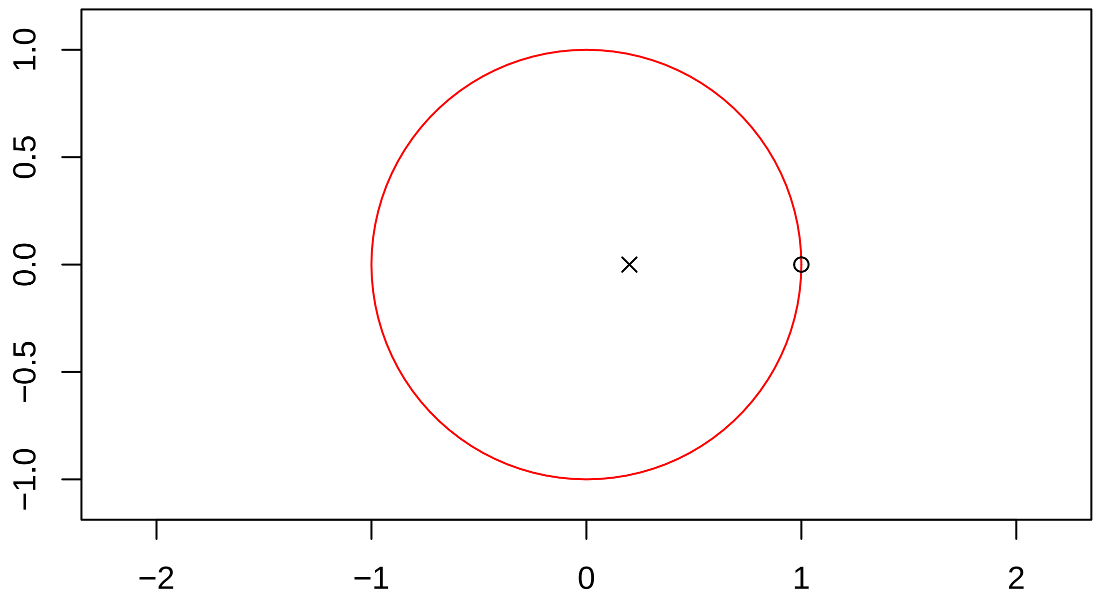

```
freqz(num1, den1)
```



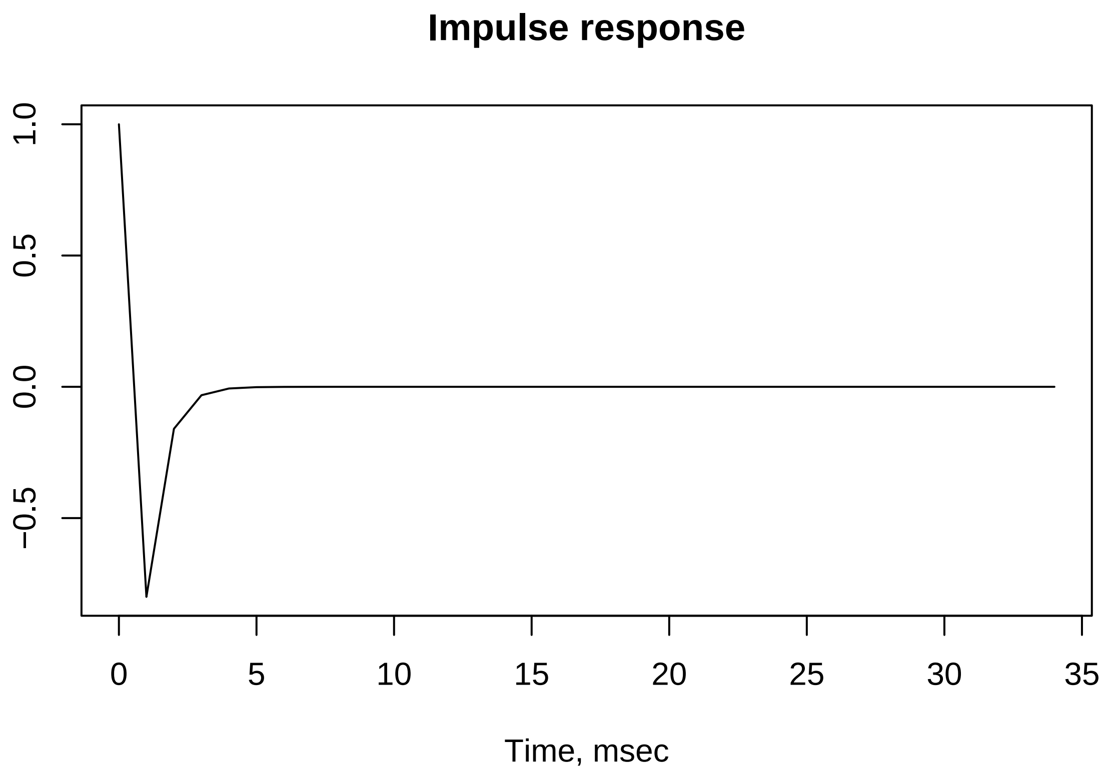

```
y2_2 <- as.data.frame(salida) # pasamos a dataframe la variable que queremos representar
indice <- c(1:nrow(y2_2)) #creamos un vector con el número de las filas que hay en
# nuestro df
y2_2 <- cbind(indice, y2_2) # unimos el vector del número de filas con la variable
# que queremos representar
y2_2 <- as.data.frame(y2_2) # lo convertimos todo a dataframe

ggplot(y2_2) + geom_line(aes(x=indice, y=x)) + xlim(0,2500) +
  ggtitle("Electrocardiograma con mu = 0.9")
```



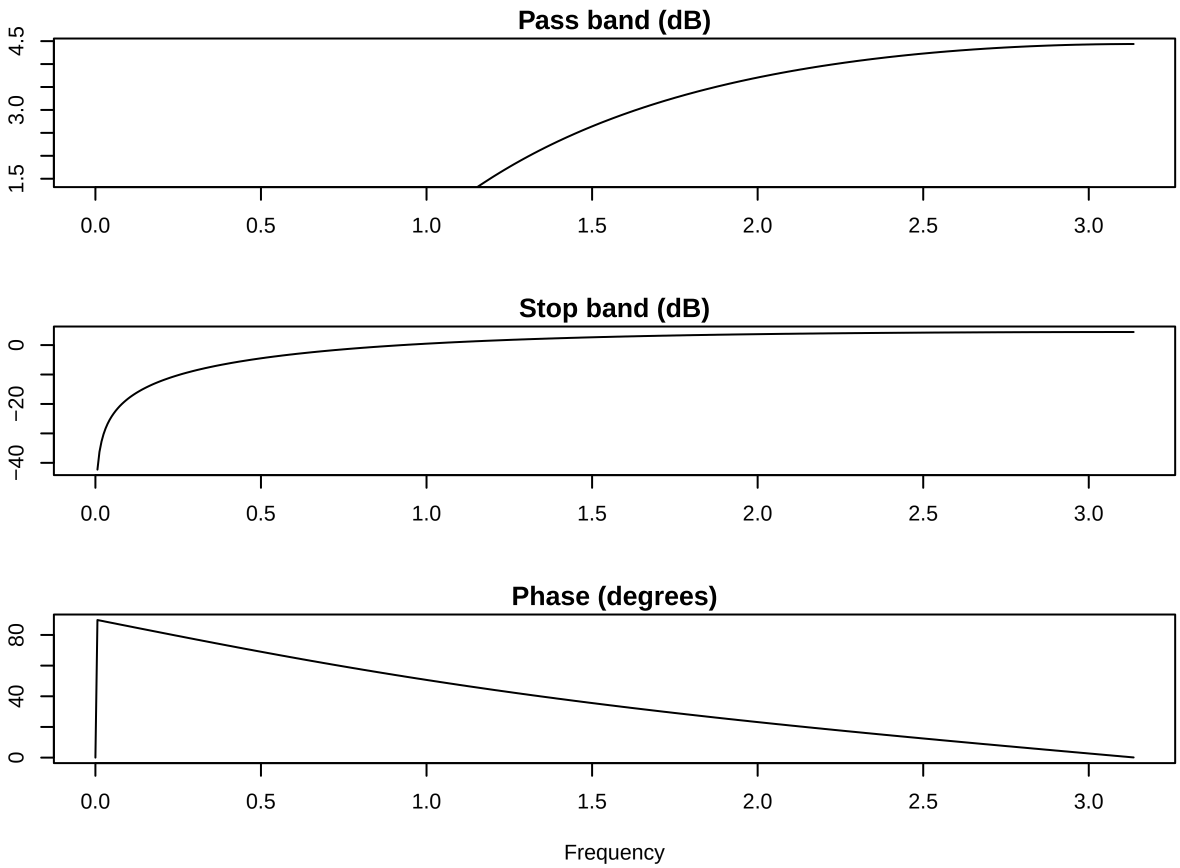

```
mu <- 0.2

num1 <- c(1,-1)
den1 <- c(1, -mu)

salida <- filter(num1, den1, p1)
zplane(num1, den1)
```



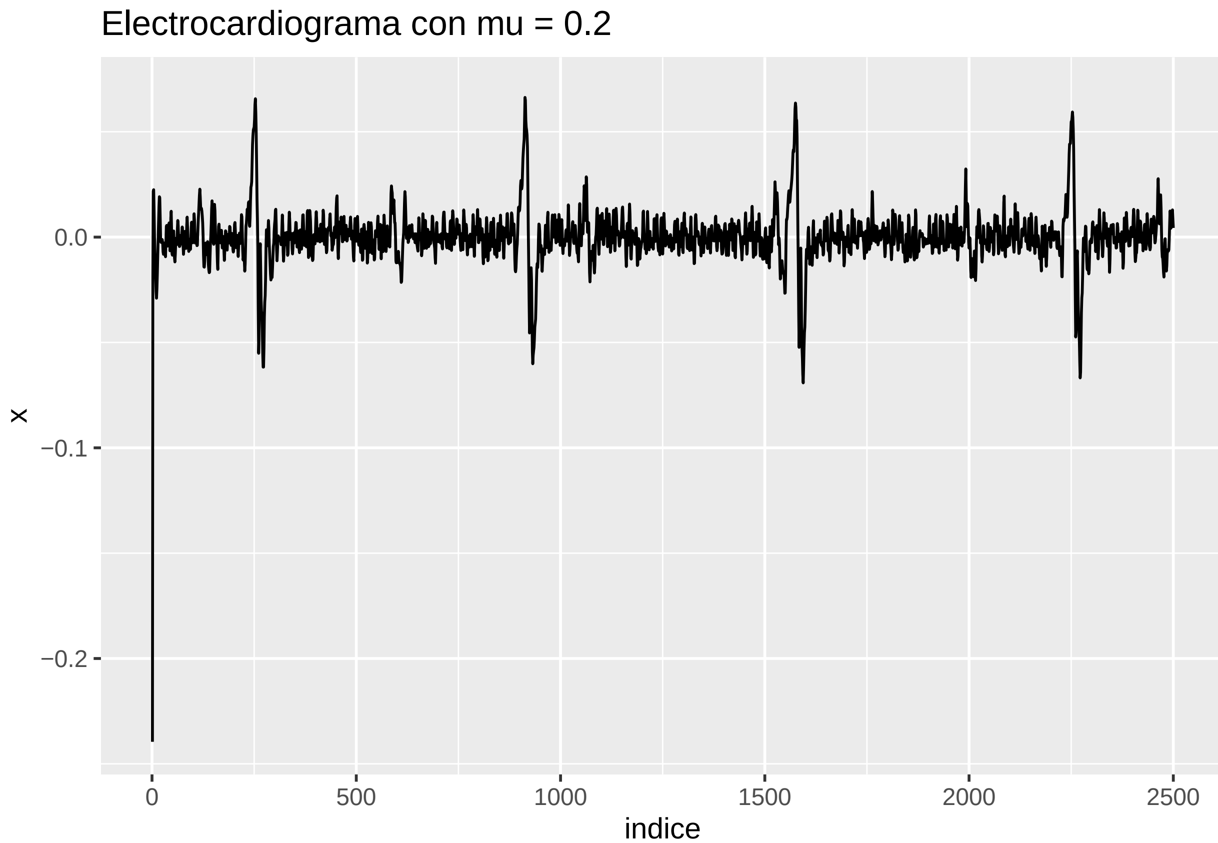

```
impz(num1, den1, 35)
```




```
freqz(num1, den1)
```




```
y2_2 <- as.data.frame(salida) # pasamos a dataframe la variable que queremos representar
indice <- c(1:nrow(y2_2)) #creamos un vector con el número de las filas que hay en
# nuestro df
y2_2 <- cbind(indice, y2_2) # unimos el vector del número de filas con la variable
# que queremos representar
y2_2 <- as.data.frame(y2_2) # lo convertimos todo a dataframe

ggplot(y2_2) + geom_line(aes(x=indice, y=x)) + xlim(0,2500) +
  ggtitle("Electrocardiograma con mu = 0.2")
```



2. Determina la respuesta en frecuencia del sistema y razona lo obtenido.

A medida que el parametro μ se va acercando al 0, la respuesta de la señal esta cada vez mas comprimida y se va acercando más al 0 como se puede comprobar graficamente con la funcion plot. Ya que en $\mu = 0.9$ como pedia el enunciado, los valores obtenidos en la respuesta rondan entre -0.3 y 0.3, mientras que con $\mu = 0.3$ los valores rondan entre -0.05 y 0.05.

3. ¿Qué controla el parámetro μ?

El parámetro mu controla los polos, puesto que esta en el denominador de la funcion H(z). Además, se puede comprobar gráficamete con la función zplane porque lo único que cambia son las cruces (polos).

Mediante el uso de la aplicación de: http://www.micromodeler.com/dsp/ visualiza los filtros que has utilizado en la práctica.