Comparacion_10_20_30-ecua_1

November 24, 2020

1 Cálculo de perfil de emisividad por métodos Nestor y Bockasten

1.0.1 A continuación por medio de expresiones regulares, extraeremos los valores de una matriz de coeficientes, esto con el objeto de poner en práctica herramientas computacionales para extracción de tablas de documentos en formato pdf, sumado a que evita errores en transcripción y escritura.

Lo que se quiere es leer los datos de la siguiente tabla encontrada en el articulo de Bockaste, para el posterior cálculo de los perfiles de emisividad.

k	$j=0$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$	$j=9$
0	+7.625972	+0.463415								
1	−5.800962	+3.606300	+0.323954							
2	−0.584698	−2.951278	+2.653847	+0.263182						
3	−0.339474	−0.182401	−2.058371	+2.198581	+0.227286					
4	−0.197038	−0.214891	−0.138728	−1.666071	+1.918418	+0.202929				
5	−0.126877	−0.134649	−0.162498	−0.112322	−1.434904	+1.723807	+0.185020			
6	−0.088278	−0.092042	−0.105026	−0.133815	−0.095626	−1.278587	+1.578512	+0.171141		
7	−0.064907	−0.066934	−0.073682	−0.087694	−0.115548	−0.084151	−1.164009	+1.464693	+0.159977	
8	−0.048250	−0.049410	−0.053181	−0.060617	−0.074289	−0.100408	−0.072617	−1.070717	+1.381857	+0.251406
9	−0.044883	−0.045711	−0.048354	−0.053365	−0.061987	−0.076986	−0.104895	−0.086465	−1.037290	+0.984158
s	9.609	4.695	3.384	2.779	2.413	2.161	1.974	1.824	1.735	1.016

k	$j=0$	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$	$j=9$
0	+15.251944	+0.926830								
1	−11.601925	+7.212600	+0.647908							
2	−1.169395	−5.902556	+5.307693	+0.526364						
3	−0.678948	−0.364801	−4.116741	+4.397163	+0.454572					
4	−0.394076	−0.429782	−0.277457	−3.332141	+3.836836	+0.405858				
5	−0.253755	−0.269299	−0.324996	−0.224644	−2.869807	+3.447613	+0.370040			
6	−0.176556	−0.184085	−0.210052	−0.267629	−0.191252	−2.557173	+3.157023	+0.342282		
7	−0.129814	−0.133868	−0.147364	−0.175388	−0.231096	−0.168301	−2.328017	+2.929387	+0.319955	
8	−0.099424	−0.101793	−0.109490	−0.124654	−0.152493	−0.205566	−0.151499	−2.150895	+2.744824	+0.301492
9	−0.078572	−0.080047	−0.084759	−0.093710	−0.109157	−0.136142	−0.186578	−0.138610	−2.008747	+2.591264
10	−0.063650	−0.064615	−0.067663	−0.073305	−0.082629	−0.097858	−0.123812	−0.171813	−0.128366	−1.891424
11	−0.052606	−0.053265	−0.055325	−0.059066	−0.065060	−0.074395	−0.089219	−0.114137	−0.159944	−0.119998
12	−0.044206	−0.044670	−0.046112	−0.048696	−0.052740	−0.058826	−0.068015	−0.082374	−0.106310	−0.150153
13	−0.037667	−0.038004	−0.039045	−0.040888	−0.043724	−0.047883	−0.053936	−0.062912	−0.076798	−0.099826
14	−0.032479	−0.032729	−0.033499	−0.034851	−0.036903	−0.039852	−0.044028	−0.049988	−0.058726	−0.072156
15	−0.028293	−0.028483	−0.029065	−0.030079	−0.031603	−0.033759	−0.036747	−0.040889	−0.046727	−0.055222
16	−0.024867	−0.025014	−0.025462	−0.026238	−0.027394	−0.029010	−0.031212	−0.034197	−0.038278	−0.043982
17	−0.022028	−0.022143	−0.022493	−0.023098	−0.023992	−0.025229	−0.026892	−0.029105	−0.032062	−0.036069
18	−0.018953	−0.019041	−0.019311	−0.019776	−0.020457	−0.021393	−0.022638	−0.024271	−0.026415	−0.029252
19	−0.020355	−0.020439	−0.020694	−0.021132	−0.021772	−0.022643	−0.023790	−0.025276	−0.027192	−0.029675
s	19.219	9.391	6.769	5.560	4.828	4.325	3.952	3.661	3.426	3.232

k	$j=10$	$j=11$	$j=12$	$j=13$	$j=14$	$j=15$	$j=16$	$j=17$	$j=18$	$j=19$
9	+0.285895									
10	+2.460896	+0.272491								
11	−1.792466	+2.348415	+0.260810							
12	−0.113010	−1.707542	+2.250070	+0.250513						
13	−0.141909	−0.107070	−1.633625	+2.163130	+0.241347					
14	−0.094349	−0.134852	−0.101947	−1.568530	+2.085548	+0.233118				
15	−0.068221	−0.089651	−0.128726	−0.097472	−1.510633	+2.015756	+0.225677			
16	−0.052239	−0.064835	−0.085566	−0.123347	−0.093522	−1.458699	+1.952533	+0.218905		
17	−0.041636	−0.049665	−0.061885	−0.081975	−0.118576	−0.090003	−1.411772	+1.894909	+0.212709	
18	−0.033075	−0.038371	−0.045999	−0.057612	−0.076732	−0.111652	−0.083135	−1.363193	+1.854467	+0.345128
19	−0.032930	−0.037283	−0.043270	−0.051835	−0.064781	−0.085922	−0.124215	−0.105395	−1.373121	+1.366258
s	3.067	2.924	2.800	2.691	2.593	2.505	2.425	2.347	2.317	1.409

[1]:

```
library("pdftools")
library("tidyverse")
library("latex2exp")
txt = pdf_text(pdf = "articulo.pdf")
txt_1 = txt[2]

#txt_1
tab <-txt_1 %>% # Remueve Todos los caracteres Que están despúés del número
  str_split("\n")
#tab
matriz = c(tab[[1]][5:17])
matriz11=as.matrix(matriz)
Ajk_n10=matrix(0,10,1)
for(i in 1:10){
  Ajk_n10[i,]=matriz11[3+i,]
}

ajk =function(r){
  as.numeric(unlist(str_extract_all(string =Ajk_n10[r,1],pattern ="[\\-]*\\d\\.
↪\\d{6,7}",simplify = T)))
}

Ajk_n102=matrix(0,10,10)
for (i in 1:10) {
  for (j in 1:10) {
    Ajk_n102[i,j]=ajk(i)[j]
  }
}
# Se corrigen algunos datos que no se pudieron leer
Ajk_n102[8,7]= -1.164009
Ajk_n102[8,8]= 1.464693
Ajk_n102[8,9]= 0.159977

print(Ajk_n102)
```

```
           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
 [1,]  7.625972  0.463415        NA        NA        NA        NA        NA
 [2,] -5.800962  3.606300  0.323954        NA        NA        NA        NA
 [3,] -0.584698 -2.951278  2.653847  0.263182        NA        NA        NA
 [4,] -0.339474 -0.182401 -2.058371  2.198581  0.227286        NA        NA
 [5,] -0.197038 -0.214891 -0.138728 -1.666071  1.918418  0.202929        NA
 [6,] -0.126877 -0.134649 -0.162498 -0.112322 -1.434904  1.723807  0.185020
 [7,] -0.088278 -0.092042 -0.105026 -0.133815 -0.095626 -1.278587  1.578512
 [8,] -0.064907 -0.066934 -0.073682 -0.087694 -0.115548 -0.084151 -1.164009
 [9,] -0.048250 -0.049410 -0.053181 -0.060617 -0.074289 -0.100408 -0.072617
[10,] -0.044883 -0.045711 -0.048354 -0.053365 -0.061987 -0.076986 -0.104895
           [,8]      [,9]     [,10]
```

```
 [1,]        NA        NA       NA
 [2,]        NA        NA       NA
 [3,]        NA        NA       NA
 [4,]        NA        NA       NA
 [5,]        NA        NA       NA
 [6,]        NA        NA       NA
 [7,]  0.171141        NA       NA
 [8,]  1.464693  0.159977       NA
 [9,] -1.070717  1.381857 0.251406
[10,] -0.086465 -1.037290 0.984158
```

Asimismo se extrajo los valores de la matriz para 20 anillos, dicha matriz esta guardada en un archivo csv, a continuación cargamos la matriz

[2]:
```
Ajk_bocka = as.matrix(read_csv(file = "Ajk_n20.csv",col_names = T))
print(Ajk_bocka)
```

```
Warning message:
‘Missing column names filled in: 'X1' [1]’Parsed with column specification:
cols(
  .default = col_double(),
  X1 = col_integer()
)
See spec(...) for full column specifications.
```

```
      X1         V1        V2        V3        V4        V5        V6        V7
 [1,]  1  15.251944  0.926830        NA        NA        NA        NA        NA
 [2,]  2 -11.601925  7.212600  0.647908        NA        NA        NA        NA
 [3,]  3  -1.169395 -5.902556  5.307693  0.526364        NA        NA        NA
 [4,]  4  -0.678948 -0.364801 -4.116741  4.397163  0.454572        NA        NA
 [5,]  5  -0.394076 -0.429782 -0.277457 -3.332141  3.836836  0.405858        NA
 [6,]  6  -0.253755 -0.269299 -0.324996 -0.224644 -2.869807  3.447613  0.370040
 [7,]  7  -0.176556 -0.184085 -0.210052 -0.267629 -0.191252 -2.557173  3.157023
 [8,]  8  -0.129814 -0.133868 -0.147364 -0.175388 -0.231096 -0.168301 -2.328017
 [9,]  9  -0.099424 -0.101793 -0.109490 -0.124654 -0.152493 -0.205566 -0.151499
[10,] 10  -0.078572 -0.080047 -0.084759 -0.093710 -0.109157 -0.136142 -0.186578
[11,] 11  -0.063650 -0.064615 -0.067663 -0.073305 -0.082629 -0.097858 -0.123812
[12,] 12  -0.052606 -0.053265 -0.055325 -0.059066 -0.065060 -0.074395 -0.089219
[13,] 13  -0.044206 -0.044670 -0.046112 -0.048696 -0.052740 -0.058826 -0.068015
[14,] 14  -0.037667 -0.038004 -0.039045 -0.040888 -0.043724 -0.047883 -0.053936
[15,] 15  -0.032479 -0.032729 -0.033499 -0.034851 -0.036903 -0.039852 -0.044028
[16,] 16  -0.028293 -0.028483 -0.029065 -0.030079 -0.031603 -0.033759 -0.036747
[17,] 17  -0.024867 -0.025014 -0.025462 -0.026238 -0.027394 -0.029010 -0.031212
[18,] 18  -0.022028 -0.022143 -0.022493 -0.023098 -0.023992 -0.025229 -0.026892
[19,] 19  -0.018953 -0.019041 -0.019311 -0.019776 -0.020457 -0.021393 -0.022638
[20,] 20  -0.020355 -0.020439 -0.020694 -0.021132 -0.021772 -0.022643 -0.023790
            V8        V9       V10       V11       V12       V13       V14
 [1,]        NA        NA        NA  0.000000  0.000000  0.000000  0.000000
 [2,]        NA        NA        NA  0.000000  0.000000  0.000000  0.000000
```

```
 [3,]        NA        NA        NA  0.000000  0.000000  0.000000  0.000000
 [4,]        NA        NA        NA  0.000000  0.000000  0.000000  0.000000
 [5,]        NA        NA        NA  0.000000  0.000000  0.000000  0.000000
 [6,]        NA        NA        NA  0.000000  0.000000  0.000000  0.000000
 [7,]  0.342282        NA        NA  0.000000  0.000000  0.000000  0.000000
 [8,]  2.929387  0.319955        NA  0.000000  0.000000  0.000000  0.000000
 [9,] -2.150895  2.744824  0.301492  0.000000  0.000000  0.000000  0.000000
[10,] -0.138610 -2.008747  2.591264  0.285895        NA        NA        NA
[11,] -0.171813 -0.128366 -1.891424  2.460896  0.272491        NA        NA
[12,] -0.114137 -0.159944 -0.119998 -1.792466  2.348415  0.260810        NA
[13,] -0.082374 -0.106310 -0.150153 -0.113010 -1.707542  2.250070  0.250513
[14,] -0.062912 -0.076798 -0.099826 -0.141909 -0.107070 -1.633625  2.163130
[15,] -0.049988 -0.058726 -0.072156 -0.094349 -0.134852 -0.101947 -1.568530
[16,] -0.040889 -0.046727 -0.055222 -0.068221 -0.089651 -0.128726 -0.097472
[17,] -0.034197 -0.038278 -0.043982 -0.052239 -0.064835 -0.085566 -0.123347
[18,] -0.029105 -0.032062 -0.036069 -0.041636 -0.049665 -0.061885 -0.081975
[19,] -0.024271 -0.026415 -0.029252 -0.033075 -0.038371 -0.045999 -0.057612
[20,] -0.025276 -0.027192 -0.029675 -0.032930 -0.037283 -0.043270 -0.051835
            V15       V16       V17       V18       V19      V20
 [1,]  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
 [2,]  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
 [3,]  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
 [4,]  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
 [5,]  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
 [6,]  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
 [7,]  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
 [8,]  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
 [9,]  0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
[10,]        NA        NA        NA        NA        NA       NA
[11,]        NA        NA        NA        NA        NA       NA
[12,]        NA        NA        NA        NA        NA       NA
[13,]        NA        NA        NA        NA        NA       NA
[14,]  0.241347        NA        NA        NA        NA       NA
[15,]  2.085548  0.233118        NA        NA        NA       NA
[16,] -1.510633  2.015756  0.225677        NA        NA       NA
[17,] -0.093522 -1.458699  1.952533  0.218905        NA       NA
[18,] -0.118576 -0.090003 -1.411772  1.894909  0.212709       NA
[19,] -0.076732 -0.111652 -0.083135 -1.363193  1.854467 0.345128
[20,] -0.064781 -0.085922 -0.124215 -0.105395 -1.373121 1.366258
```

1.0.2 Inicialmente se variará el número de anillos, viendo la influencia de estos en los gráficos de error.

Crearemos la función error que principalmente encontrará la diferencia entre de la función de emisividad teórica con la función de emisividad numérica (emisividad encontrada con la matriz de Nestor).

[3]:

```r
library(tidyverse)
library(ggplot2)
library("latex2exp")

error = function(Anillos){


    aA_c = matrix(0,Anillos,Anillos)
    p =seq(0,Anillos-1)
    q =seq(0,Anillos-1)

    Ajk_h =function(j,k){
      (sqrt(abs((j+1)^2-k^2))-sqrt(j^2-k^2))/(2*j+1)
    }

    Ajk_men1h =function(j,k){
      (sqrt(abs(j^2-k^2))-sqrt((j-1)^2-k^2))/(2*j-1)
    }





    suppressWarnings(for (m in p) {
      for (l in q) {
        ifelse(m==l, {aA_c[l+1,m+1]=-Ajk_h(l,m)},{
 ↪aA_c[l+1,m+1]=Ajk_men1h(l,m)-Ajk_h(l,m)})
      }
    })
    aA_c


    ################################ Calculo grafica
 ↪################################



    r= rep(0,Anillos)
    r_0 = 1
    for (i in 1:{Anillos-1}) {
      r[i+1]=r[i]+r_0/Anillos
    }
    #r[8]=r[8]+0.001 #errores relacionados cona la función epsilon_3
    #r[10]=r[10]+.00001
```

```
    e_num = matrix(0,1,Anillos)
    #e_num_b = matrix(0,1,Anillos)

    #G1 = (1-r^4)
    G1 = 1600*(1-r^2)
    #e_teo = (4/(3*pi*r_0))*(1+2*r^2)*sqrt(1-r^2)
    e_teo =(3200/pi)*sqrt(1-r^2)
    e_teo

    for (i in 1:Anillos-1) {
      #e_num[i]=(-1/(pi*r_0))*sum(aA_c[,i]*G1,na.rm = TRUE) #na.rm = TRUE Ignor␣
 ↪los valores los NA's
      e_num[i]=(-2/((r_0/Anillos)*pi))*sum(aA_c[,i]*G1,na.rm = TRUE)
      #e_num_b[i]=sum(Ajk_n20[,i]*G1,na.rm = TRUE)
    }
   return(e_num)
    #as.numeric(abs(e_num-e_teo))
}
```

1.1 A continuación coeficientes $a_{j,k}$ calculados con el articulo de Nestor para 20 Anillos.

[4]:
```
Anillos = 20
deci=7
aA_c = matrix(0,Anillos,Anillos)
    p =seq(0,Anillos-1)
    q =seq(0,Anillos-1)

    Ajk_h =function(j,k){
      (sqrt(abs((j+1)^2-k^2))-sqrt(j^2-k^2))/(2*j+1)
    }

    Ajk_men1h =function(j,k){
      (sqrt(abs(j^2-k^2))-sqrt((j-1)^2-k^2))/(2*j-1)
    }






    suppressWarnings(for (m in p) {
      for (l in q) {
        ifelse(m==l, {aA_c[l+1,m+1]=round(-Ajk_h(l,m),deci)},{␣
 ↪aA_c[l+1,m+1]=round(Ajk_men1h(l,m)-Ajk_h(l,m),deci)})
      }
    })
```

```r
print(aA_c)
```

```
           [,1]       [,2]       [,3]       [,4]       [,5]       [,6]
 [1,] -1.0000000        NaN        NaN        NaN        NaN        NaN
 [2,]  0.6666667 -0.5773503        NaN        NaN        NaN        NaN
 [3,]  0.1333333  0.3580750 -0.4472136        NaN        NaN        NaN
 [4,]  0.0571429  0.0700529  0.2717802 -0.3779645        NaN        NaN
 [5,]  0.0317460  0.0352227  0.0511585  0.2274924 -0.3333333        NaN
 [6,]  0.0202020  0.0215359  0.0266132  0.0417309  0.1995028 -0.3015113
 [7,]  0.0139860  0.0146080  0.0167886  0.0219409  0.0359516  0.1797918
 [8,]  0.0102564  0.0105856  0.0116889  0.0140240  0.0189696  0.0319850
 [9,]  0.0078431  0.0080337  0.0086549  0.0098890  0.0122003  0.0168924
[10,]  0.0061920  0.0063099  0.0066875  0.0074077  0.0086621  0.0108980
[11,]  0.0050125  0.0050895  0.0053325  0.0057836  0.0065328  0.0077677
[12,]  0.0041408  0.0041931  0.0043569  0.0046548  0.0051335  0.0058828
[13,]  0.0034783  0.0035151  0.0036295  0.0038347  0.0041564  0.0046421
[14,]  0.0029630  0.0029896  0.0030720  0.0032181  0.0034432  0.0037739
[15,]  0.0025543  0.0025741  0.0026349  0.0027419  0.0029044  0.0031384
[16,]  0.0022247  0.0022397  0.0022856  0.0023659  0.0024863  0.0026571
[17,]  0.0019550  0.0019666  0.0020020  0.0020633  0.0021547  0.0022824
[18,]  0.0017316  0.0017407  0.0017683  0.0018161  0.0018866  0.0019844
[19,]  0.0015444  0.0015516  0.0015735  0.0016113  0.0016667  0.0017428
[20,]  0.0013860  0.0013918  0.0014094  0.0014397  0.0014838  0.0015440
           [,7]       [,8]       [,9]      [,10]      [,11]      [,12]
 [1,]        NaN        NaN        NaN        NaN        NaN        NaN
 [2,]        NaN        NaN        NaN        NaN        NaN        NaN
 [3,]        NaN        NaN        NaN        NaN        NaN        NaN
 [4,]        NaN        NaN        NaN        NaN        NaN        NaN
 [5,]        NaN        NaN        NaN        NaN        NaN        NaN
 [6,]        NaN        NaN        NaN        NaN        NaN        NaN
 [7,] -0.2773501        NaN        NaN        NaN        NaN        NaN
 [8,]  0.1649533 -0.2581989        NaN        NaN        NaN        NaN
 [9,]  0.0290614  0.1532653 -0.2425356        NaN        NaN        NaN
[10,]  0.0153461  0.0267981  0.1437517 -0.2294157        NaN        NaN
[11,]  0.0099157  0.0141425  0.0249823  0.1358130 -0.2182179        NaN
[12,]  0.0070840  0.0091446  0.0131739  0.0234854  0.1290582 -0.2085144
[13,]  0.0053793  0.0065421  0.0085206  0.0123742  0.0222248  0.1232195
[14,]  0.0042567  0.0049765  0.0061007  0.0080033  0.0117003  0.0211448
[15,]  0.0034703  0.0039455  0.0046460  0.0057331  0.0075663  0.0111230
[16,]  0.0028940  0.0032231  0.0036885  0.0043694  0.0054213  0.0071912
[17,]  0.0024567  0.0026933  0.0030176  0.0034722  0.0041339  0.0051531
[18,]  0.0021158  0.0022910  0.0025253  0.0028436  0.0032873  0.0039306
[19,]  0.0018440  0.0019769  0.0021514  0.0023825  0.0026943  0.0031271
[20,]  0.0016234  0.0017263  0.0018593  0.0020321  0.0022593  0.0025645
          [,13]      [,14]      [,15]      [,16]      [,17]      [,18]
 [1,]        NaN        NaN        NaN        NaN        NaN        NaN
 [2,]        NaN        NaN        NaN        NaN        NaN        NaN
```

```
 [3,]        NaN        NaN        NaN        NaN        NaN        NaN
 [4,]        NaN        NaN        NaN        NaN        NaN        NaN
 [5,]        NaN        NaN        NaN        NaN        NaN        NaN
 [6,]        NaN        NaN        NaN        NaN        NaN        NaN
 [7,]        NaN        NaN        NaN        NaN        NaN        NaN
 [8,]        NaN        NaN        NaN        NaN        NaN        NaN
 [9,]        NaN        NaN        NaN        NaN        NaN        NaN
[10,]        NaN        NaN        NaN        NaN        NaN        NaN
[11,]        NaN        NaN        NaN        NaN        NaN        NaN
[12,]        NaN        NaN        NaN        NaN        NaN        NaN
[13,] -0.2000000        NaN        NaN        NaN        NaN        NaN
[14,]  0.1181073 -0.1924501        NaN        NaN        NaN        NaN
[15,]  0.0202066  0.1135824 -0.1856953        NaN        NaN        NaN
[16,]  0.0106214  0.0193817  0.1095404 -0.1796053        NaN        NaN
[17,]  0.0068650  0.0101807  0.0186493  0.1059012 -0.1740777        NaN
[18,]  0.0049193  0.0065781  0.0097896  0.0179934  0.1026020 -0.1690309
[19,]  0.0037530  0.0047134  0.0063234  0.0094396  0.0174016  0.0995930
[20,]  0.0029868  0.0035963  0.0045304  0.0060954  0.0091241  0.0168642
          [,19]      [,20]
 [1,]       NaN        NaN
 [2,]       NaN        NaN
 [3,]       NaN        NaN
 [4,]       NaN        NaN
 [5,]       NaN        NaN
 [6,]       NaN        NaN
 [7,]       NaN        NaN
 [8,]       NaN        NaN
 [9,]       NaN        NaN
[10,]       NaN        NaN
[11,]       NaN        NaN
[12,]       NaN        NaN
[13,]       NaN        NaN
[14,]       NaN        NaN
[15,]       NaN        NaN
[16,]       NaN        NaN
[17,]       NaN        NaN
[18,]       NaN        NaN
[19,] -0.164399        NaN
[20,]  0.096834 -0.1601282
```

A continuación crearemos la función radioo que reproducirá un vector con diferentes radios, la enunciada función esta en tiene como parametro x el numero de anillos, por tanto si introduzco radioo(10) tomará diviciones desde cero(0) hasta uno (1) con avances de 0.1.

[5]:
```
radioo=function(x){
    rr =rep(0,x)
    r_0 = 1
    for (i in 1:{x-1}) {
```

```
        rr[i+1]=rr[i]+r_0/x
    }
as.vector(rr)
 }
```

[6]:
```
radioo(10)
```

1. 0 2. 0.1 3. 0.2 4. 0.3 5. 0.4 6. 0.5 7. 0.6 8. 0.7 9. 0.8 10. 0.9

Usando las funciones creadas anteriormente (error y radioo), se crea un data.frame que contendrá una columna que almacene el número de anillos con el que se calculo el error, en otra columna los valores de error, y en otra el radio, esto con el objeto de comparar los gráficos con diferente número de anillos.

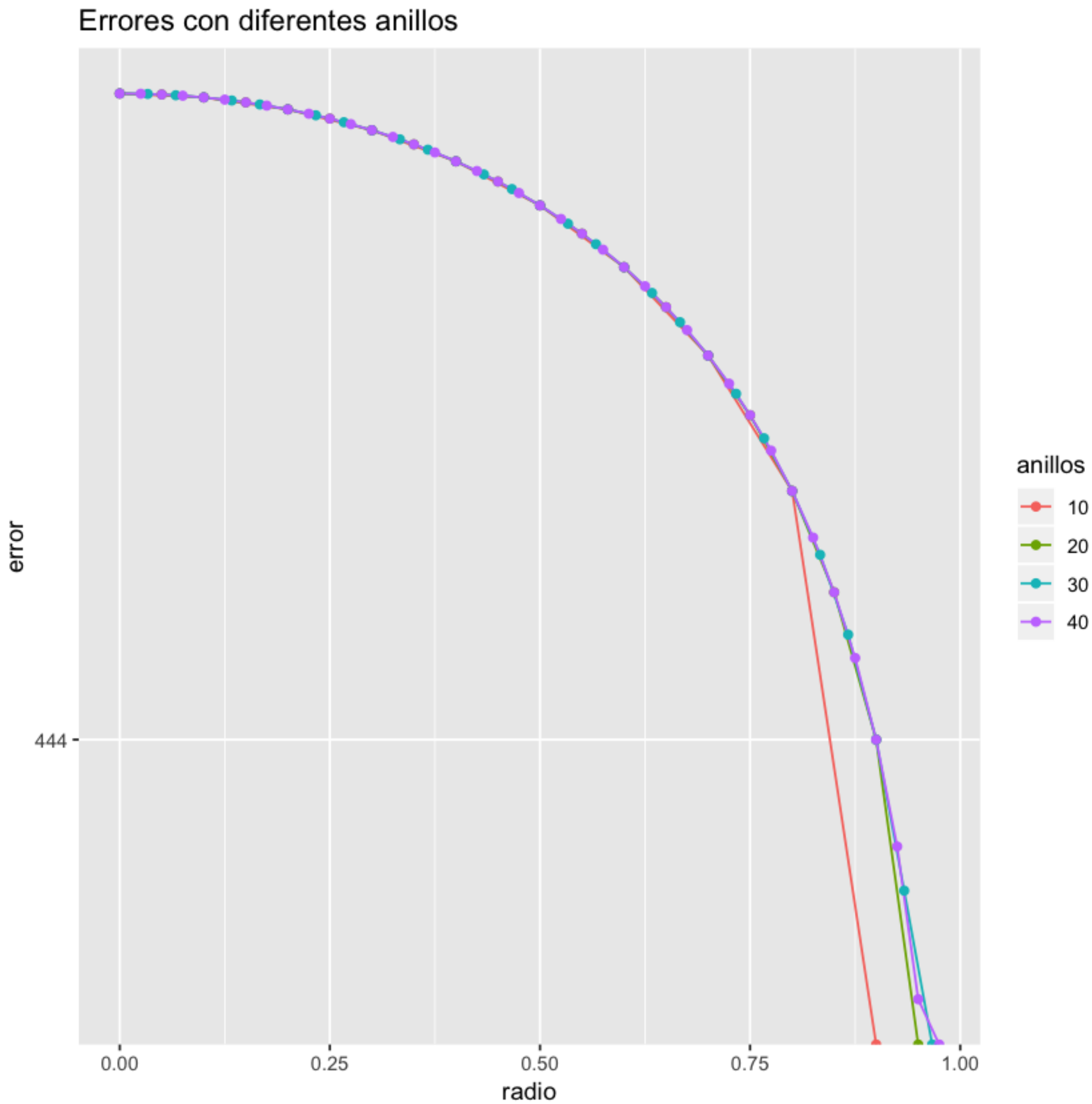
[7]:
```
err_10 = error(10)
error_10 = list(anillos=c(rep("10",10)),
                error = as.vector(err_10),
                radio =as.vector(radioo(10)) )
err_20 = error(20)
error_20 = list(anillos=c(rep("20",20)),
                error = as.vector(err_20),
                radio =as.vector(radioo(20)) )

err_30 = error(30)
error_30 = list(anillos=c(rep("30",30)),
                error = as.vector(err_30),
                radio =as.vector(radioo(30)) )
err_40 = error(40)
error_40 = list(anillos=c(rep("40",40)),
                error = as.vector(err_40),
                radio =as.vector(radioo(40)) )
```

[8]:
```
errores = bind_rows(error_10,error_20,error_30,error_40)
#errores=errores%>%filter(radio<0.87)

a=0.0000000000001
b = 0.0000000000001
errores%>%
  ggplot(aes(radio,error,color = anillos))+
  geom_line()+
  geom_point()+
  ggtitle("Errores con diferentes anillos")+
  #scale_y_continuous(trans = "log10",breaks =c(0,1,10,100,1000))
  scale_y_continuous(trans = "log10",breaks =c(a,0.5*a,3*a,10*a,20*a,444))

max(errores$error)
```

```
Warning message:
```

```
“Transformation introduced infinite values in continuous y-axis”Warning message:
“Transformation introduced infinite values in continuous y-axis”
```

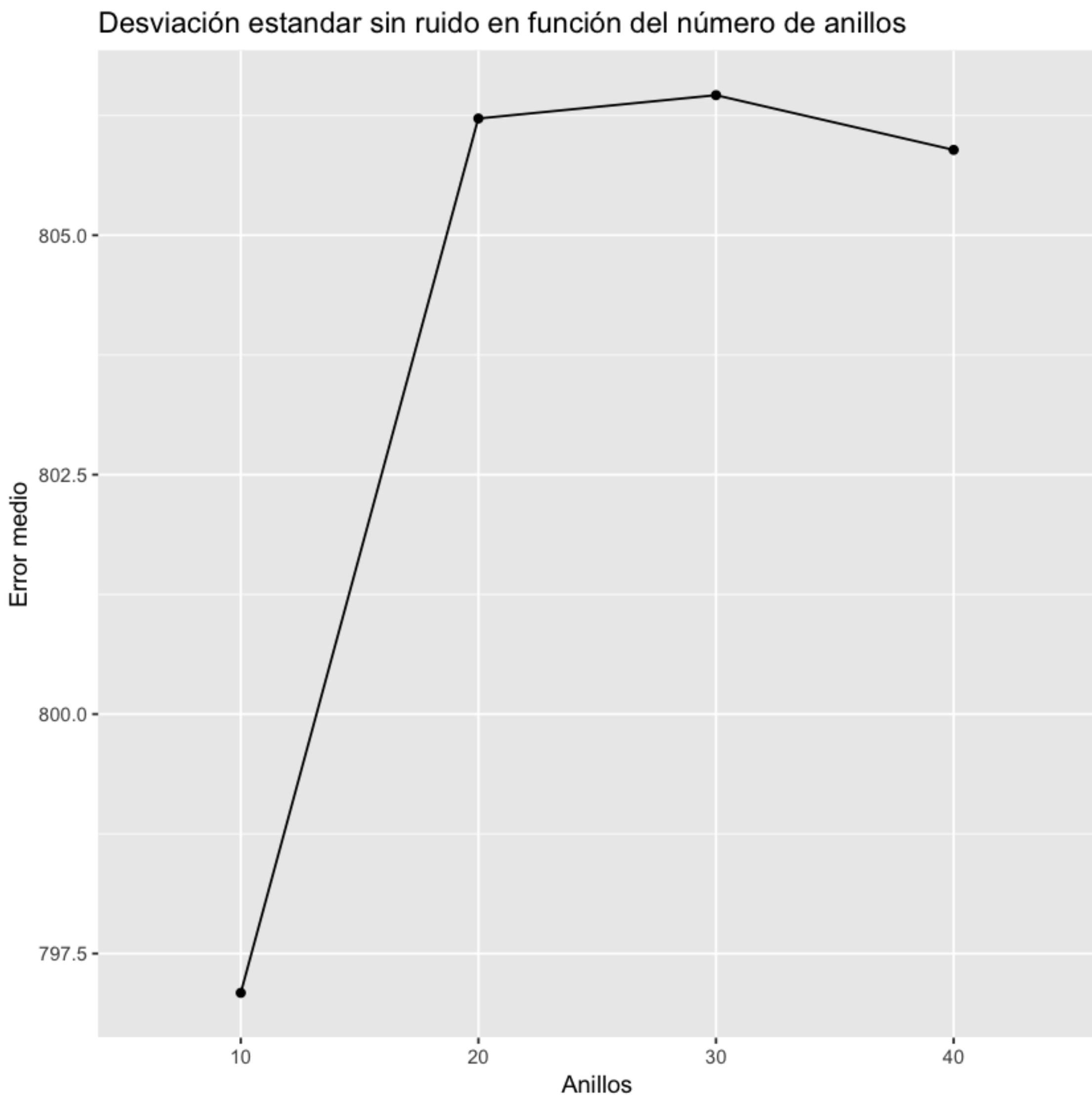
1018.59163578813



Por el momento solo he creado objetos que contienen información de cada método o número de anillos, ahora vamos a unir el todo esto en solo objeto llamado errores, para poder graficarlos y ver las diferencias entre tomar 10,20,30 o 40 anillos.

[9]:
```
errores%>%group_by(anillos)%>%
          #filter(radio<0.95)%>%
          summarize(error_medio=mean(error))%>%
          ggplot(aes(anillos,error_medio, group = 1))+geom_point()+geom_line()+
```

```
        labs(title ="Desviación estandar sin ruido en función del número de␣
↪anillos",
        y=quote('Error medio'),
        x=quote(Anillos))
```



[10]:
```
#rrores%>%group_by(anillos)
errores%>%filter(anillos==10)%>%summarize(n())#summarise(error_medio=sum(error)/
↪n())
#           summarize(error_medio=mean(error))
```

n()
10

1.1.1 Procederemos a variar el número de cifras decimales para corroborar si hay cambios significativos en las gráficas de error.

Creamos la función **error_deci** con el objeto de variar número de anillos y decimales tomados, para corroborar como varía el error en función de las cifras decimales tomadas.

[11]:
```
error_deci = function(Anillos,deci){



    aA_c = matrix(0,Anillos,Anillos)
    p =seq(0,Anillos-1)
    q =seq(0,Anillos-1)

    Ajk_h =function(j,k){
      (sqrt(abs((j+1)^2-k^2))-sqrt(j^2-k^2))/(2*j+1)
    }

    Ajk_men1h =function(j,k){
      (sqrt(abs(j^2-k^2))-sqrt((j-1)^2-k^2))/(2*j-1)
    }






    suppressWarnings(for (m in p) {
      for (l in q) {
        ifelse(m==l, {aA_c[l+1,m+1]=round(-Ajk_h(l,m),deci)},{
  aA_c[l+1,m+1]=round(Ajk_men1h(l,m)-Ajk_h(l,m),deci)})
      }
    })
    aA_c

    r= rep(0,Anillos)
    r_0 = 1
    for (i in 1:{Anillos-1}) {
      r[i+1]=r[i]+r_0/Anillos
    }

    e_num = matrix(0,1,Anillos)

    G1 = 1600*(1-r^2)
    e_teo =(3200/pi)*sqrt(1-r^2)
    e_teo

    for (i in 1:Anillos-1) {
      e_num[i]=(-2/((r_0/Anillos)*pi))*sum(aA_c[,i]*G1,na.rm = TRUE)
    }
```

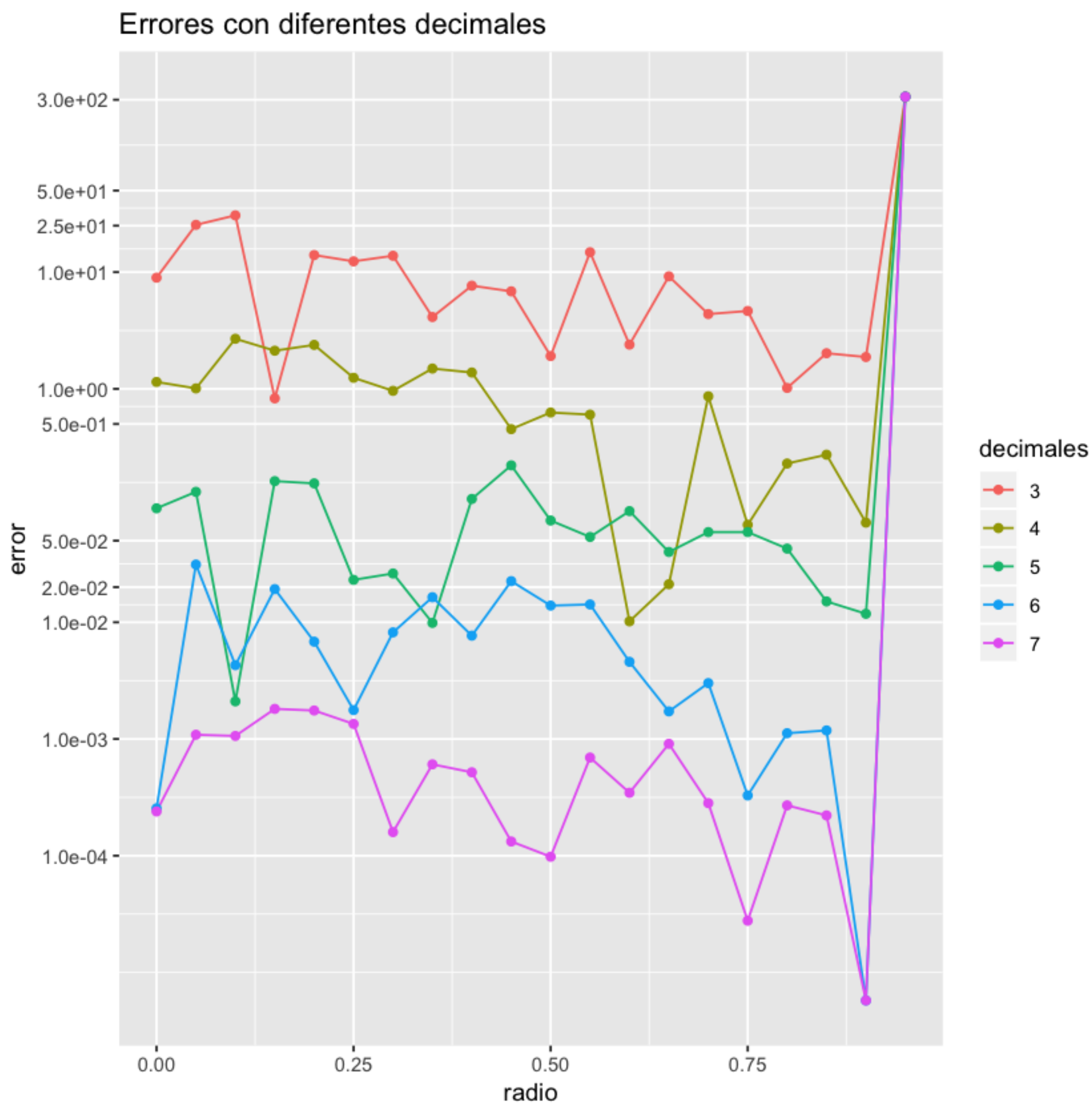
```
e_num
    as.numeric(abs(e_num-e_teo))

}
```

A continuación se crean los objetos que contienen información dependiendo de los decimales tomados y del número de anillos, de la misma manera una columna que indique cuantos decimales se estan tomando a consideración, usando de referencia el método que implementa el cálculo para 20 anillos.

[12]:
```
err_20_3_deci = error_deci(20,3)
error_20_3_deci = list(decimales=c(rep("3",20)),
                error = as.vector(err_20_3_deci),
                radio =as.vector(radioo(20)) )
err_20_4_deci = error_deci(20,4)
error_20_4_deci = list(decimales=c(rep("4",20)),
                error = as.vector(err_20_4_deci),
                radio =as.vector(radioo(20)) )
err_20_5_deci = error_deci(20,5)
error_20_5_deci = list(decimales=c(rep("5",20)),
                error = as.vector(err_20_5_deci),
                radio =as.vector(radioo(20)) )
err_20_6_deci = error_deci(20,6)
error_20_6_deci = list(decimales=c(rep("6",20)),
                error = as.vector(err_20_6_deci),
                radio =as.vector(radioo(20)) )
err_20_7_deci = error_deci(20,7)
error_20_7_deci = list(decimales=c(rep("7",20)),
                error = as.vector(err_20_7_deci),
                radio =as.vector(radioo(20)) )
```

Unimos con la función **bind_rows** los objetos y graficamos, veamos como la linea de la escala me deja por medio de la función **ggplot2** modificar los valores que quiero que me muestre en las ordenadas por medio de un vector, introduciendolo en el argumento breaks.

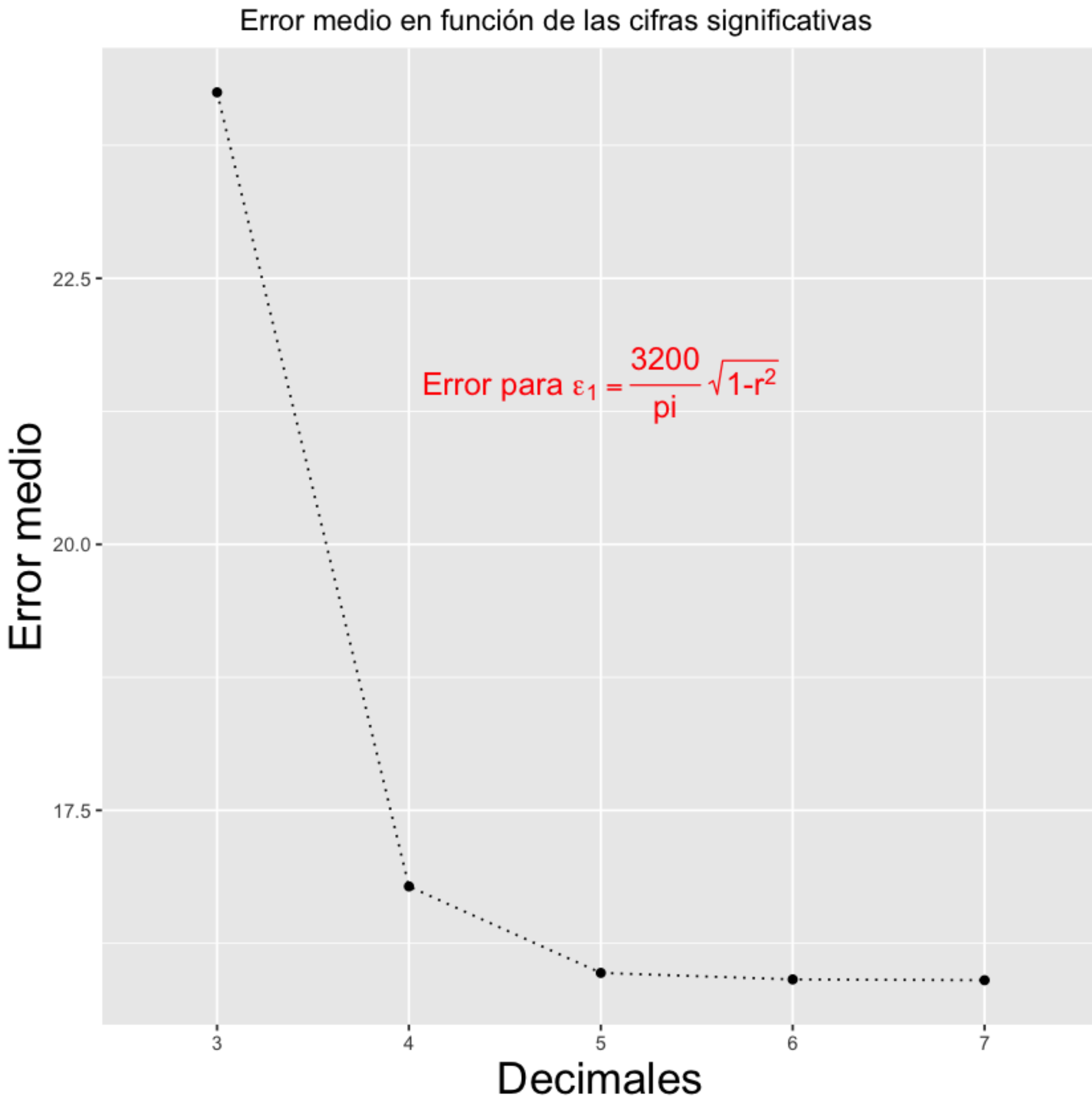
[13]:
```
errores_deci345 =␣
↪bind_rows(error_20_3_deci,error_20_4_deci,error_20_5_deci,error_20_6_deci,error_20_7_deci)


errores_deci345%>%
  ggplot(aes(radio,error,color = decimales))+
  geom_line()+
  geom_point()+
  ggtitle("Errores con diferentes decimales")+
  scale_y_continuous(trans = "log2",breaks = c(0.0001,0.001,0.01,0.02,0.05,0.
↪5,1,10,25,50,300))
```



[14]:
```
errores_deci345%>%group_by(decimales)%>%
          #filter(radio<0.95)%>%
          summarize(e_medio=mean(error))%>%
          ggplot(aes(decimales,e_medio, group =␣
 ↪1))+geom_point()+geom_line(linetype="dotted")+
          labs(title ="                   Error medio en función de las cifras␣
 ↪significativas",
          y=quote("Error medio"),
          x=quote('Decimales'))+
          theme(#axis.text=element_text(size=13))#,
              axis.title.y=element_text(size=20),
              axis.title.x=element_text(size=20))+
```

```
            annotate(geom="text", x=3, y=21.5, label=TeX("Error para␣
↪$\\epsilon_{1}=\\frac{3200}{pi}\\,\\sqrt{1-r^2}$"),color="red",size=5)
```

```
Warning message in is.na(x):
“is.na() aplicado a un objeto que no es (lista o vector) de tipo 'expression'”
```



1.1.2 Ahora compararemos para el uso del método con 20 anillos las dos diferentes formas aplicadas por los autores, uno en el artículo de Bokasten y lo otro en el artículo de Nestor.

A continuación se llamaran los valores de la matriz para 20 anillos que se encuentra en el articulo de Bockasten, esto se hace por medio de un objeto con extención **.csv**, se calculan valores de emisividades con las dos diferentes matrices, una que se encuentra enunciada en el articulo de Bockasten, y la otra en el articulo de Nestor que espeficifica el método exacto para replicarla y también poder reproducir el método con diferentes números de anillos.

Ya habiendo cargado nuestra matriz podremos calcular los valores de emisividad, y comparar con valores de radio para 20 anillos. A continuación se encuentra el valor numérico y el teórico para la emisividad ε y se guarda el valor absoluto de la diferencia de estos dos valores en el objeto **error_b**.

[15]:
```
e_num_b = matrix(0,1,20)

G1b = 1600*(1-radioo(20)^2)
e_teob =(3200/pi)*sqrt(1-radioo(20)^2)


for (i in 1:20) {
      e_num_b[i]=sum(Ajk_bocka[,i+1]*G1b,na.rm = TRUE)
    }
error_b =as.numeric(abs(e_num_b-e_teob))
```

[16]:
```
e_num_b
```

1018.592 1017.317 1013.487 1007.068 998.0134 986.2471 971.6733 954.1645 933.5545 909.6308 8

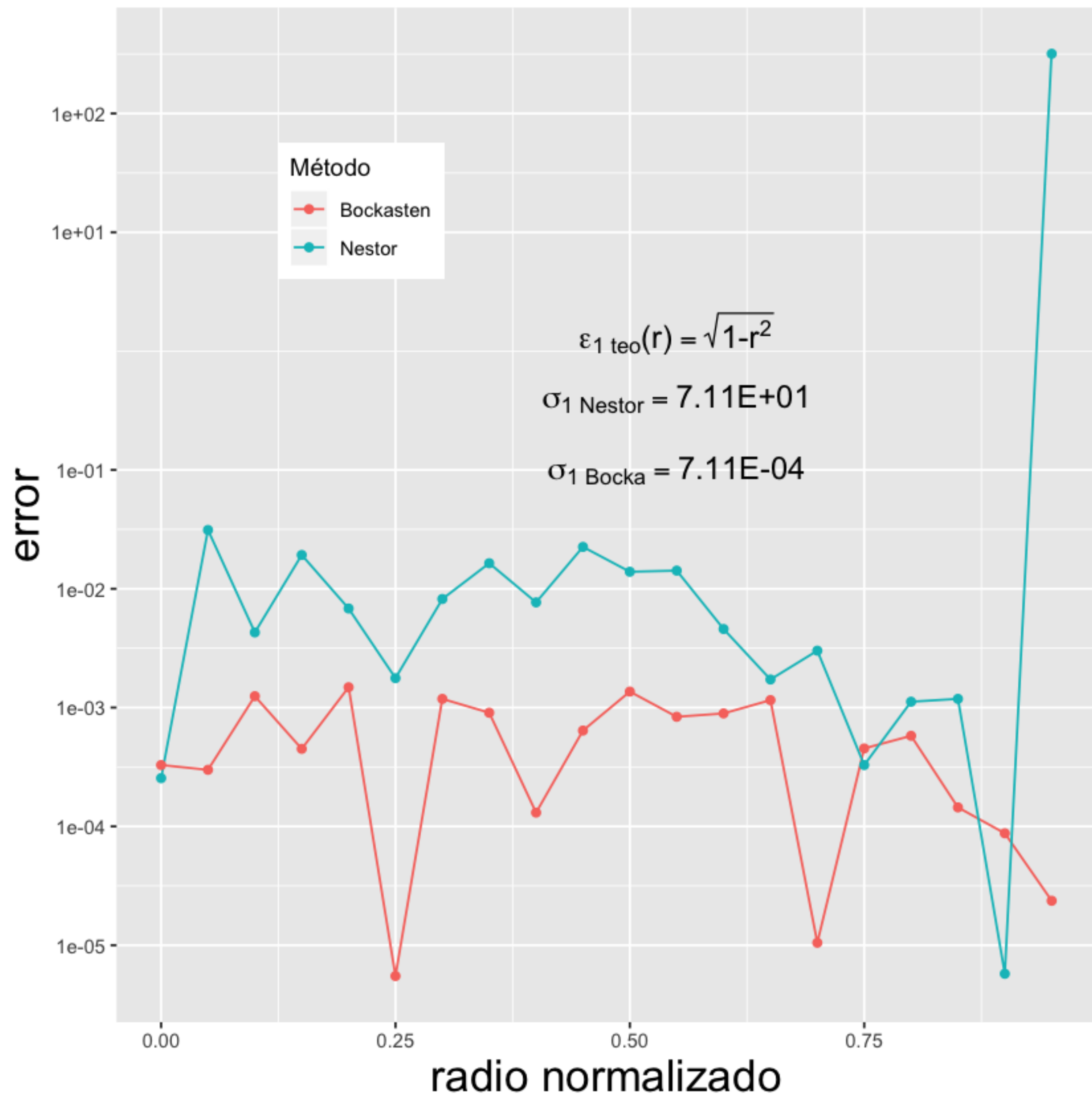
Crearemos los objetos que enseguida seran unidos con la columna que me indicará el valor de error asociado a el tipo de método usado, para luego graficar la diferencia entre métodos.

[17]:
```
err_20_6_deci = error_deci(20,6)
error_20_nest = list(Método=c(rep("Nestor",20)),
                error = as.vector(err_20_6_deci),
                radio =as.vector(radioo(20)) )



error_20_bock = list(Método=c(rep("Bockasten",20)),
                error = as.vector(error_b),
                radio =as.vector(radioo(20)))
```

[18]:
```
errores_BN = bind_rows(error_20_nest,error_20_bock)



errores_BN%>%
  ggplot(aes(radio,error,color = Método))+
  geom_line()+
  geom_point()+
  #scale_y_continuous("log()")
  scale_y_continuous(trans = "log10",breaks␣
 ↪=c(10^3,10^2,10^1,10^-1,10^-2,10^-3,10^-4,10^-5))+
  labs(    x="radio normalizado",
          y="error")+
          theme(#axis.text=element_text(size=13))#,
          axis.title.y=element_text(size=20),
```

```
        axis.title.x=element_text(size=20),)+
    theme(legend.position=c(.25, 0.8))+
    annotate(geom="text", x=.55, y=1.5,␣
↪label=TeX("$\\epsilon_{1\\,teo}(r)=\\sqrt{1-r^{2}}$"),color="black",size=5)+
    annotate(geom="text", x=.55, y=0.4, label=TeX("$\\sigma_{1\\,Nestor}= 7.
↪11E+01$"),color="black",size=5)+
    annotate(geom="text", x=.55, y=0.1, label=TeX("$\\sigma_{1\\,Bocka}= 7.
↪11E-04$"),color="black",size=5)

############

   r= rep(0,20)
   r_0 = 1
   for (i in 1:{20-1}) {
     r[i+1]=r[i]+r_0/20
   }

   e_teo =(3200/pi)*sqrt(1-r^2)





###########
```

```
Warning message in is.na(x):
“is.na() aplicado a un objeto que no es (lista o vector) de tipo
'expression'”Warning message in is.na(x):
“is.na() aplicado a un objeto que no es (lista o vector) de tipo
'expression'”Warning message in is.na(x):
“is.na() aplicado a un objeto que no es (lista o vector) de tipo 'expression'”
```



[19]:
```
deci=6
Anillos=20
g_anill = function(nn){
    integrales=integrales[,nn]%>%filter(.!=0 )%>%pull(.)         ########
 ↪g_anill    ########
    }

    aA_c = matrix(0,Anillos,Anillos)
    p =seq(0,Anillos-1)
    q =seq(0,Anillos-1)

    Ajk_h =function(j,k){
      (sqrt(abs((j+1)^2-k^2))-sqrt(j^2-k^2))/(2*j+1)
```

```
    }

    Ajk_men1h =function(j,k){
      (sqrt(abs(j^2-k^2))-sqrt((j-1)^2-k^2))/(2*j-1)
    }






    suppressWarnings(for (m in p) {
      for (l in q) {
        ifelse(m==l, {aA_c[l+1,m+1]=round(-Ajk_h(l,m),deci)},{
↪aA_c[l+1,m+1]=round(Ajk_men1h(l,m)-Ajk_h(l,m),deci)})
      }
    })
   # aA_c

    r= rep(0,Anillos)
    r_0 = 1
    for (i in 1:{Anillos-1}) {
      r[i+1]=r[i]+r_0/Anillos
    }

    e_num = matrix(0,1,Anillos)

    G1 = 1600*(1-r^2)
    #e_teo = (4/(3*pi*r_0))*(1+2*r^2)*sqrt(1-r^2)
    e_teo =(3200/pi)*sqrt(1-r^2)

    for (i in 1:Anillos-1) {
      e_num[i]=(-2/((r_0/Anillos)*pi))*sum(aA_c[,i]*G1,na.rm = TRUE)
    }

    #as.numeric(abs(e_num-e_teo))
print(e_num_b)

u = e_teo
e_k_b = e_num_b
e_k_n = e_num
s.d.nestor = as.numeric(sqrt((1/20)*sum((e_k_n-u)^2)))
print(s.d.nestor)
s.d.bocka = as.numeric(sqrt((1/20)*sum((e_k_b-u)^2)))
print(s.d.bocka)
```

```
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]     [,7]     [,8]
[1,] 1018.592 1017.317 1013.487 1007.068 998.0134 986.2471 971.6733 954.1645
```

```
         [,9]    [,10]    [,11]    [,12]    [,13]    [,14]    [,15]    [,16]
[1,] 933.5545 909.6308 882.1276 850.6926 814.8724 774.0615 727.4199 673.7355
        [,17]    [,18]    [,19]    [,20]
[1,] 611.1556 536.5766 443.9939 318.0552
[1] 71.11929
[1] 0.0007764969
```