

Algoritmo_1

November 23, 2020

- 1 Con la función $g_1(r) = 1600(1 - r^2)$ normalizada, agregaremos ruido a $g(r)$ a travez de la función sample.

```
[20]: library(tidyverse)
load("coeficientes_nestor.RData")
library("latex2exp")
##### Ruido variable #####
Anillos = 30
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 = (1-r^2)
e_teo =(2/pi)*sqrt(1-r^2)

for (i in 1:Anillos) {
  e_num[i]=(-2/((r_0/Anillos)*pi))*sum(aA_c[,i]*G1,na.rm = TRUE)
}
error=as.numeric(abs(e_num-e_teo))
ruido =10
set.seed(42)
G1_con_ruido =(sample(seq(-ruido*.01,ruido*.01,0.01),Anillos,replace = TRUE))+G1

##### Ruido variable #####

e_num_rui = matrix(0,1,Anillos)
e_teo_ruido =(2/pi)*sqrt(1-r^2)

for (i in 1:Anillos) {
  e_num_rui[i]=(-2/((r_0/Anillos)*pi))*sum(aA_c[,i]*G1_con_ruido,na.rm = TRUE)
}
```

```

error_rui=as.numeric(abs(e_num_rui-e_teo_ruido))

error_sin_ruido = list(Descripción=c(rep("Error sin ruido",30)),
                        err = as.vector(error),
                        radio = r )
error_con_ruido = list(Descripción =c(rep("Error con ruido",30)),
                        err =as.vector(error_rui),
                        radio = r)

G1_sin_ruido = list(Descripción=c(rep("g sin ruido",30)),
                    g = as.vector(G1),
                    radio = r )
G1_ruido = list(Descripción =c(rep("g con ruido",30)),
                g =as.vector(G1_con_ruido),
                radio = r)

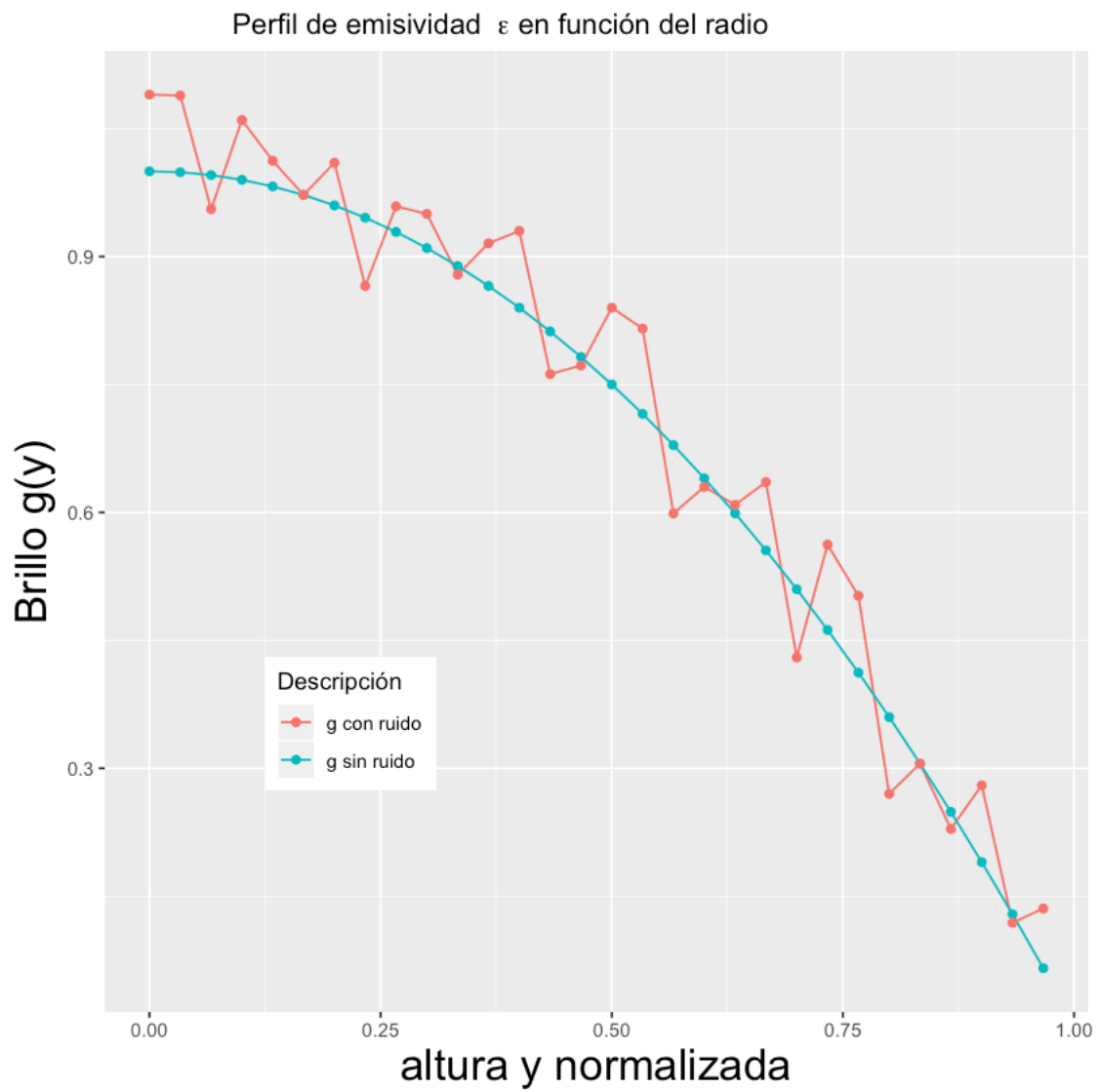
Gs = bind_rows(G1_sin_ruido,G1_ruido)
Gs%>%
  ggplot(aes(radio,g,color = Descripción))+
  geom_line()+
  geom_point()+
  labs(title=expression("Perfil de emisividad " *epsilon* " en
  ↪función del radio"),
        x="altura y normalizada",
        y=TeX("Brillo g(y)"))+
  theme(
    axis.title.y=element_text(size=20),
    axis.title.x=element_text(size=20),)+
  theme(legend.position=c(.25, 0.3))

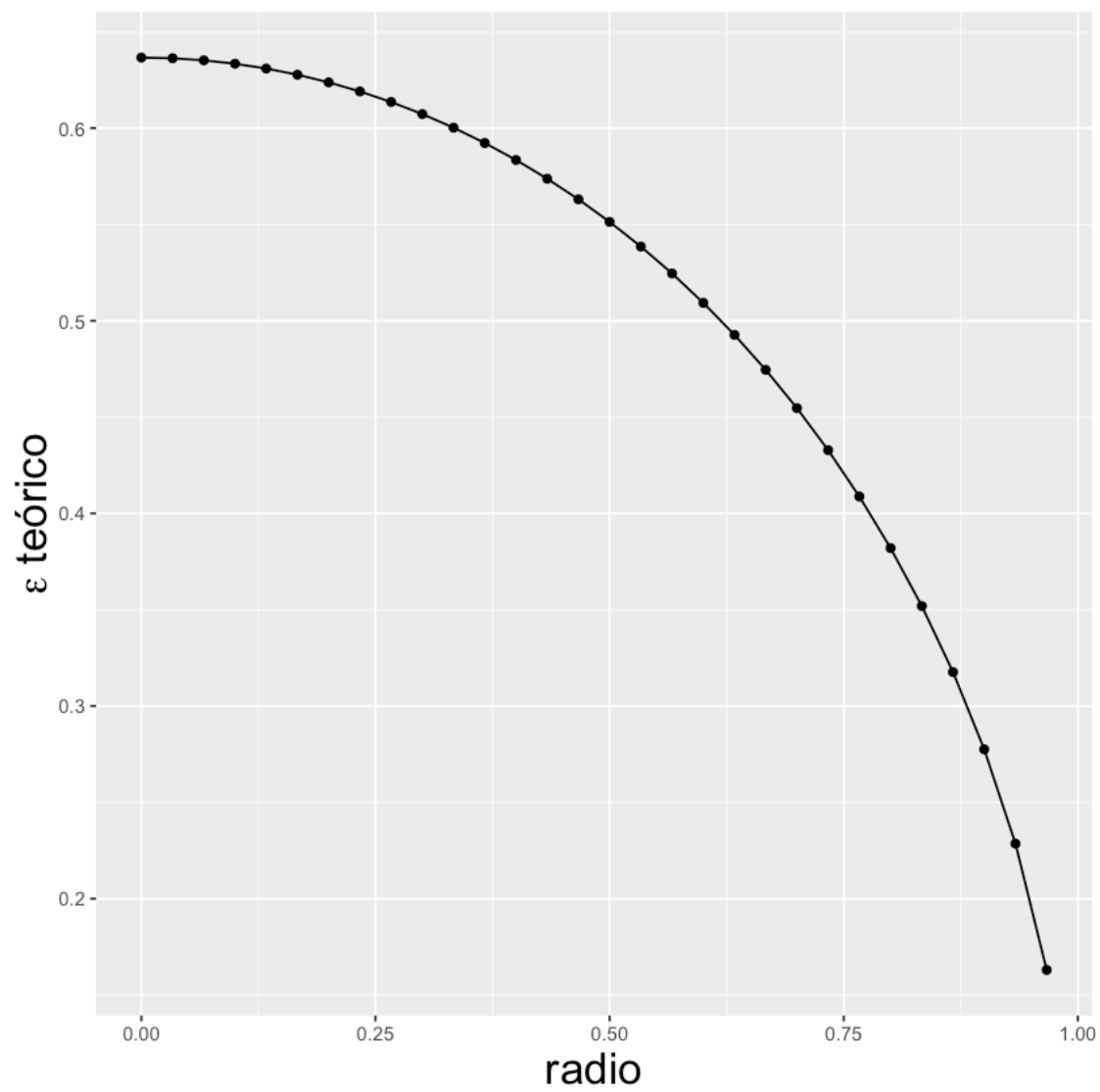
ggplot()+
  geom_line(aes(as.vector(r),as.vector(e_teo)))+
  geom_point(aes(as.vector(r),as.vector(e_teo)))+
  labs(
    x="radio",
    y=TeX("$\\epsilon$, teórico$"))+
  theme(
    axis.title.y=element_text(size=20),
    axis.title.x=element_text(size=20),)

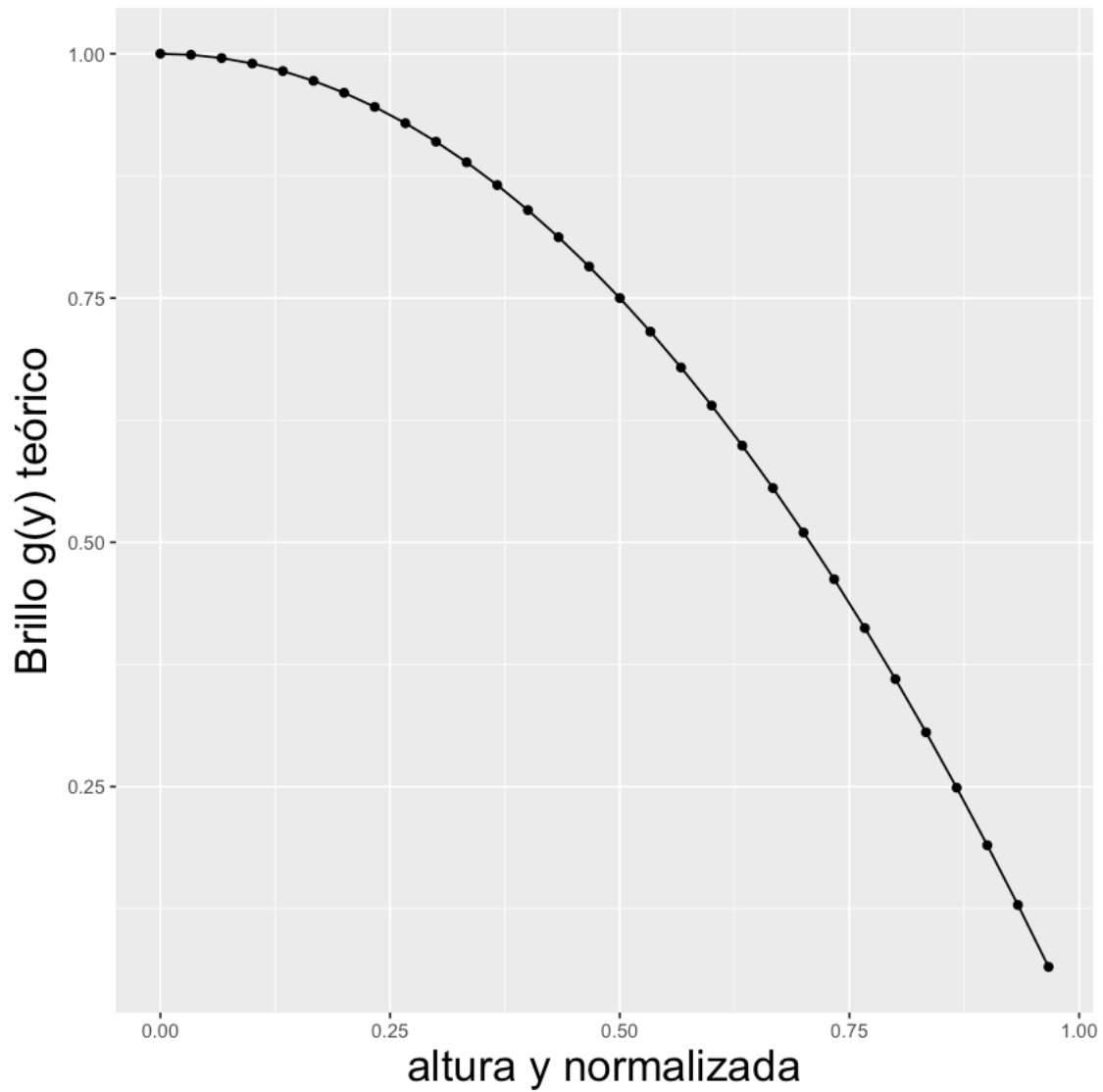
ggplot()+
  geom_line(aes(as.vector(r),as.vector(G1)))+
  geom_point(aes(as.vector(r),as.vector(G1)))+
  labs(
    x="altura y normalizada",
    y=TeX("Brillo g(y) teórico"))+

```

```
theme(
  axis.title.y=element_text(size=20),
  axis.title.x=element_text(size=20),)#+
```







```
[2]: install.packages("latex2exp")
library("latex2exp")
labels=data.frame(método=c("Método teórico","Método numérico"),x=c(0.3,0.
↪6),y=c(0.4,0.8))
aa = list(Descripción=c(rep("Perfil teórico",30)),
          g = as.vector(e_num),
          radio = r )
bb = list(Descripción =c(rep("Perfil numérico",30)),
          g =as.vector(e_teo),
          radio = r)

Gs = bind_rows(aa,bb)
Gs%>%ggplot(aes(radio,g,col=Descripción))+
```

```

geom_point(size = 2.5, stroke = 0, shape = 16)+
geom_line(size = 0.4)+
labs(title=expression("
↪en función del radio"),
      x="radio de columna de plasma normalizado",
      y=expression(" emisividad "*epsilon))+
theme(
  axis.title.y=element_text(size=20),
  axis.title.x=element_text(size=20),)+
theme(legend.position=c(.25, 0.3))+
annotate(geom="text", x=.75, y=0.55,↪
↪label=TeX("$\\epsilon_1=\\sqrt{1-r^2}$"),color="lightseagreen",size=5)

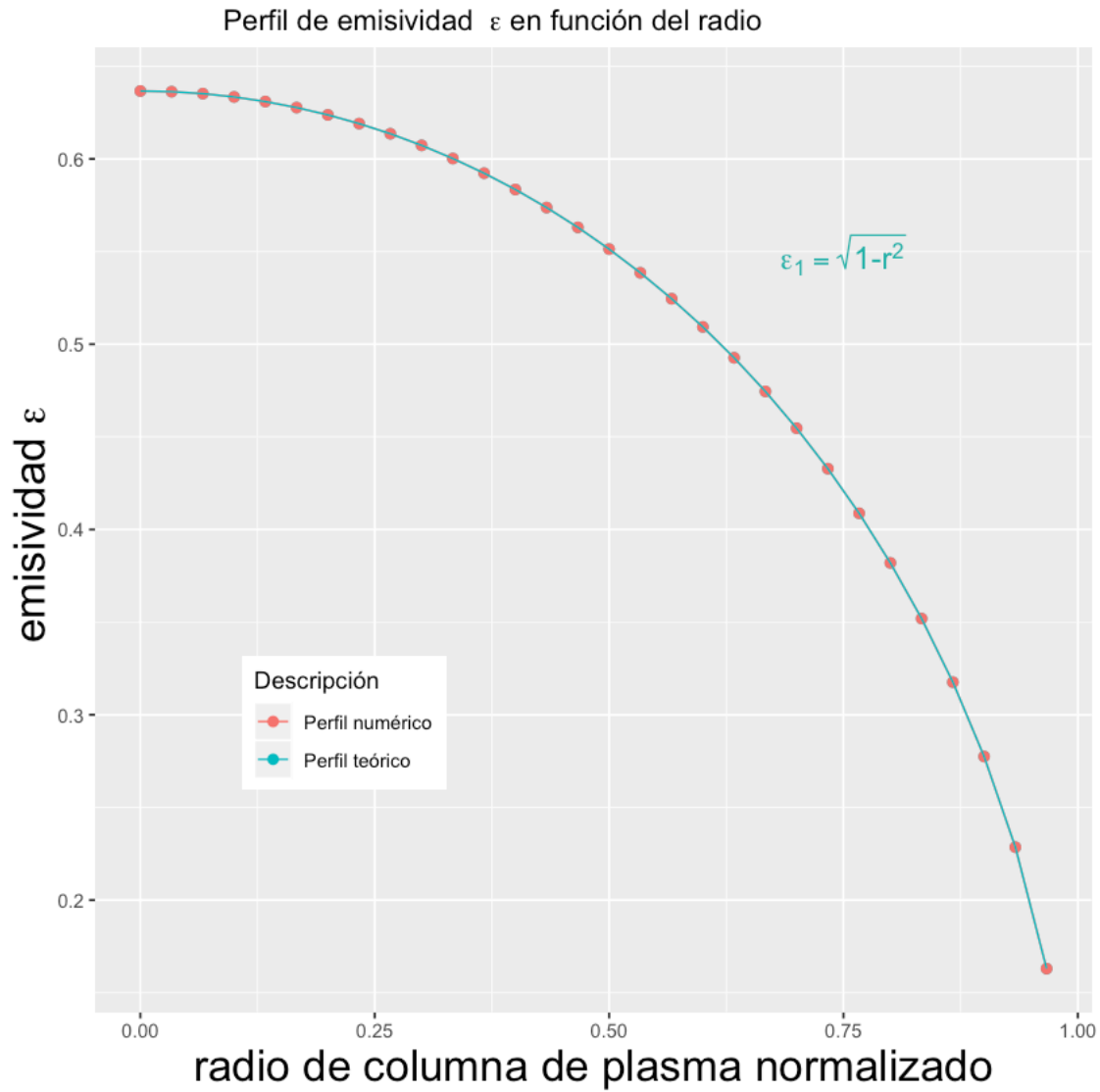
```

Updating HTML index of packages in '.Library'

Making 'packages.html' ... done

Warning message in is.na(x):

"is.na() aplicado a un objeto que no es (lista o vector) de tipo 'expression'"



Graficamos los errores con ruido, y sin ruido.

```
[4]: u = e_teo
e_k = e_num
e_k.r = e_num_rui
ds = as.numeric(sqrt((1/Anillos)*sum((e_k-u)^2)))
ds.r = as.numeric(sqrt((1/Anillos)*sum((e_k.r-u)^2)))
ds.r
ds
```

0.37394717834613

4.05396129632541e-16

```
[5]: ruiditos = bind_rows(error_sin_ruido,error_con_ruido)
ruiditos%>%
  ggplot(aes(radius,err,color = Descripción))+
  geom_line()+
  geom_point()+
  ggtitle("                Perfil de error")+
  scale_y_continuous(trans = "log10")+
  #scale_y_continuous(limit = c(0,1))
  #scale_y_continuous(limit = c(0,0.0000000000000001))+
  ylab("error")+
  labs(#title=expression("                Perfil de emisividad  "*epsilon*" en
  →función del radio"),
        x="radio",
        y=TeX("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, 0.3))#+
  annotate(geom="text", x=.25, y=10^-9, label=TeX("$\\sigma_{1 ruido}=0.
  →37$"),color="red",size=5)+
  annotate(geom="text", x=.3, y=10^-10, label=TeX("$\\sigma_{1 sin
  →ruido}=4.05x(10^{-16})$"),color="lightseagreen",size=5)+
  annotate(geom="text", x=.3, y=10^-6,
  →label=TeX("$error\\,para\\,\\epsilon_1=\\sqrt{1-r^2}$"),color="black",size=4)
```

Warning message:

"Transformation introduced infinite values in continuous y-axis"Warning message:

"Transformation introduced infinite values in continuous y-axis"Warning message
in is.na(x):

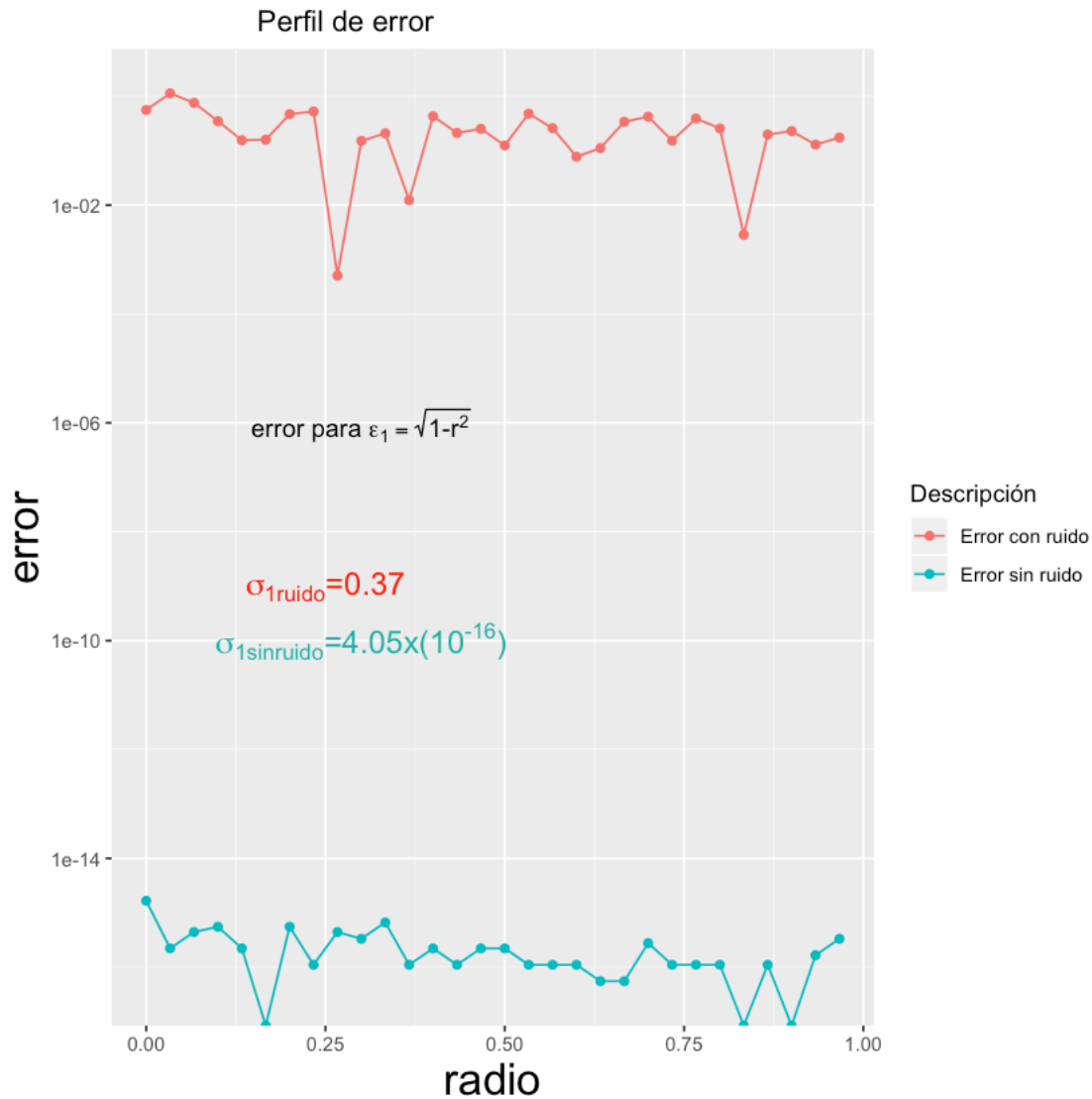
"is.na() aplicado a un objeto que no es (lista o vector) de tipo

'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"

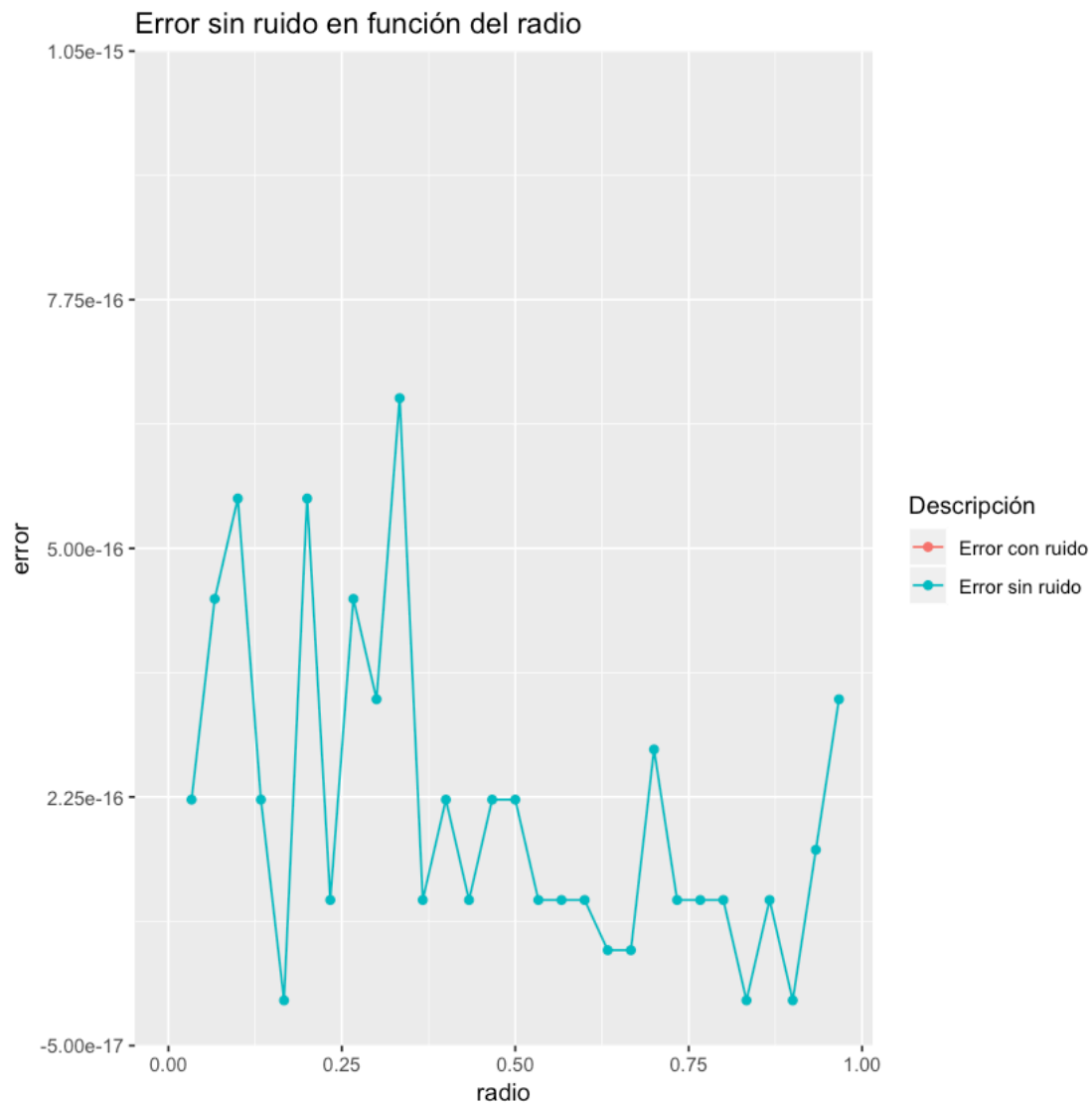


a continuación ampliaremos la anterior gráfica en verde, para ver que el error cuando la función $g(r)$ no tiene ruido no es totalmente cero, pero tampoco fluctúan tanto sus valores.

```
[6]: ruiditos%>%
  ggplot(aes(radio,err,color = Descripción))+
  geom_line()+
  geom_point()+
  ggtitle("Error sin ruido en función del radio")+
  #scale_y_continuous(trans = "log10")+
  #scale_y_continuous(limit = c(0,1))
  scale_y_continuous(limit = c(0,0.0000000000000001))+
  ylab("error")
```

Warning message:

"Removed 31 rows containing missing values (geom_path)."
 Warning message:
 "Removed 31 rows containing missing values (geom_point)."



```
[7]: library(tidyverse)
ds_diego = 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==1, {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 con 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 = (1-r^2)
#e_teo = (4/(3*pi*r_0))*(1+2*r^2)*sqrt(1-r^2)
e_teo =(2/pi)*sqrt(1-r^2)
e_teo

for (i in 1:Anillos) {
  #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)
  }
  u = e_teo
  e_k = e_num
  as.numeric(sqrt((1/Anillos)*sum((e_k-u)^2)))
}

```

```

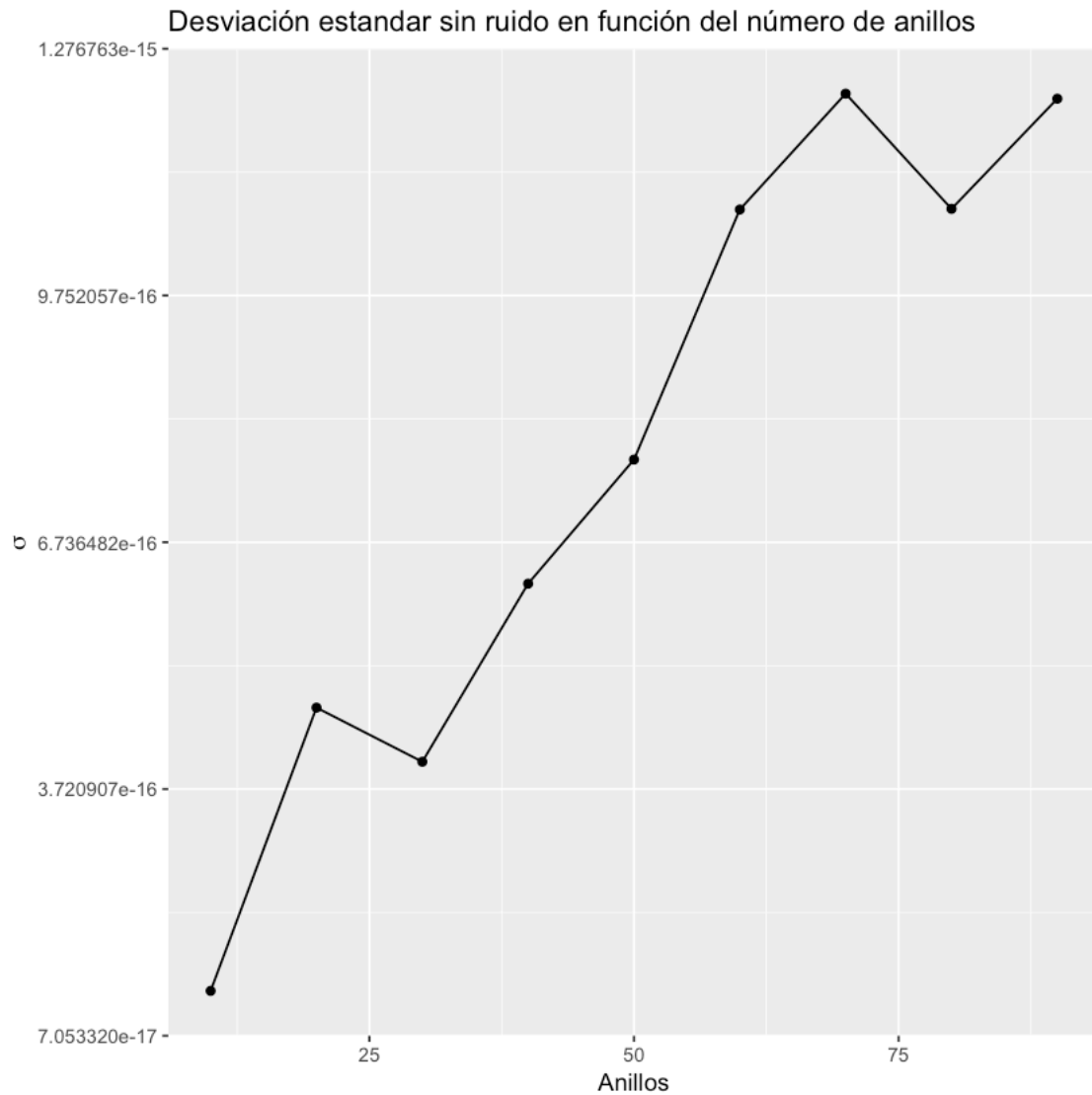
[8]: ani = seq(10,90,10)
     ds =sapply(ani,ds_diego)
     d_frame_ds = as.tibble(ds)%>%mutate(ani=ani)

```

```

[9]: library(ggplot2)
     d_frame_ds%>%ggplot(aes(ani,ds))+
     geom_point()+
     geom_line()+
     labs(title ="Desviación estandar sin ruido en función del número de anillos",
           y=quote(sigma),
           x=quote(Anillos))

```



Ahora a funciones con un ruido del 10% se analizara la dependencia de su desviación estandar, en función de la cantidad de anillos.

```
[10]: ds_diego_ruido = 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==1, {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

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

e_num_rui = matrix(0,1,Anillos)

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

##### Implementación del ruido a la funcion #####

set.seed(42)
G1_con_ruido =(sample(seq(-0.1,0.1,0.01),Anillos,replace = TRUE))+G1
e_num_rui = matrix(0,1,Anillos)

for (i in 1:Anillos) {
  e_num_rui[i]=(-2/((r_0/Anillos)*pi))*sum(aA_c[,i]*G1_con_ruido,na.rm = TRUE)
}
u = e_teo
e_k =e_num_rui
as.numeric(sqrt((1/Anillos)*sum((e_k-u)^2)))
}

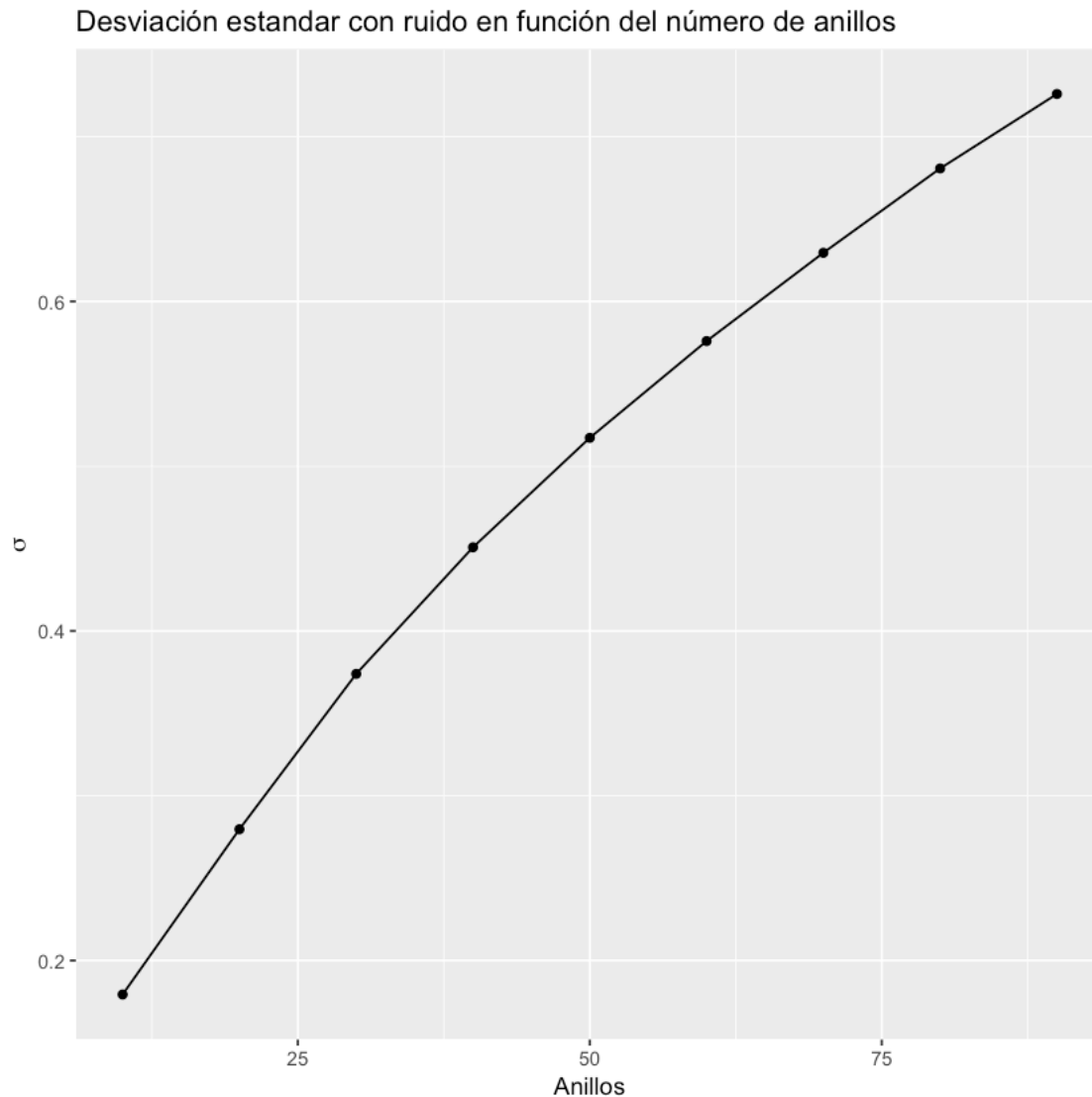
```

```
#####
#ds_ruido =sapply(ani,ds_diego_ruido)
#d_frame_ds_ruido = as.tibble(ds_ruido)%>%mutate(ani=ani)

#d_frame_ds_ruido%>%ggplot(aes(ani,ds))+
#geom_point()+
#geom_line()
```

```
[11]: ds_ruido =sapply(ani,ds_diego_ruido)
d_frame_ds_ruido = as.tibble(ds_ruido)%>%mutate(ani=ani)

d_frame_ds_ruido%>%ggplot(aes(ani,ds_ruido))+
geom_point()+
geom_line()+
  labs(title ="Desviación estandar con ruido en función del número de anillos",
        y=quote(sigma),
        x=quote(Anillos))
```

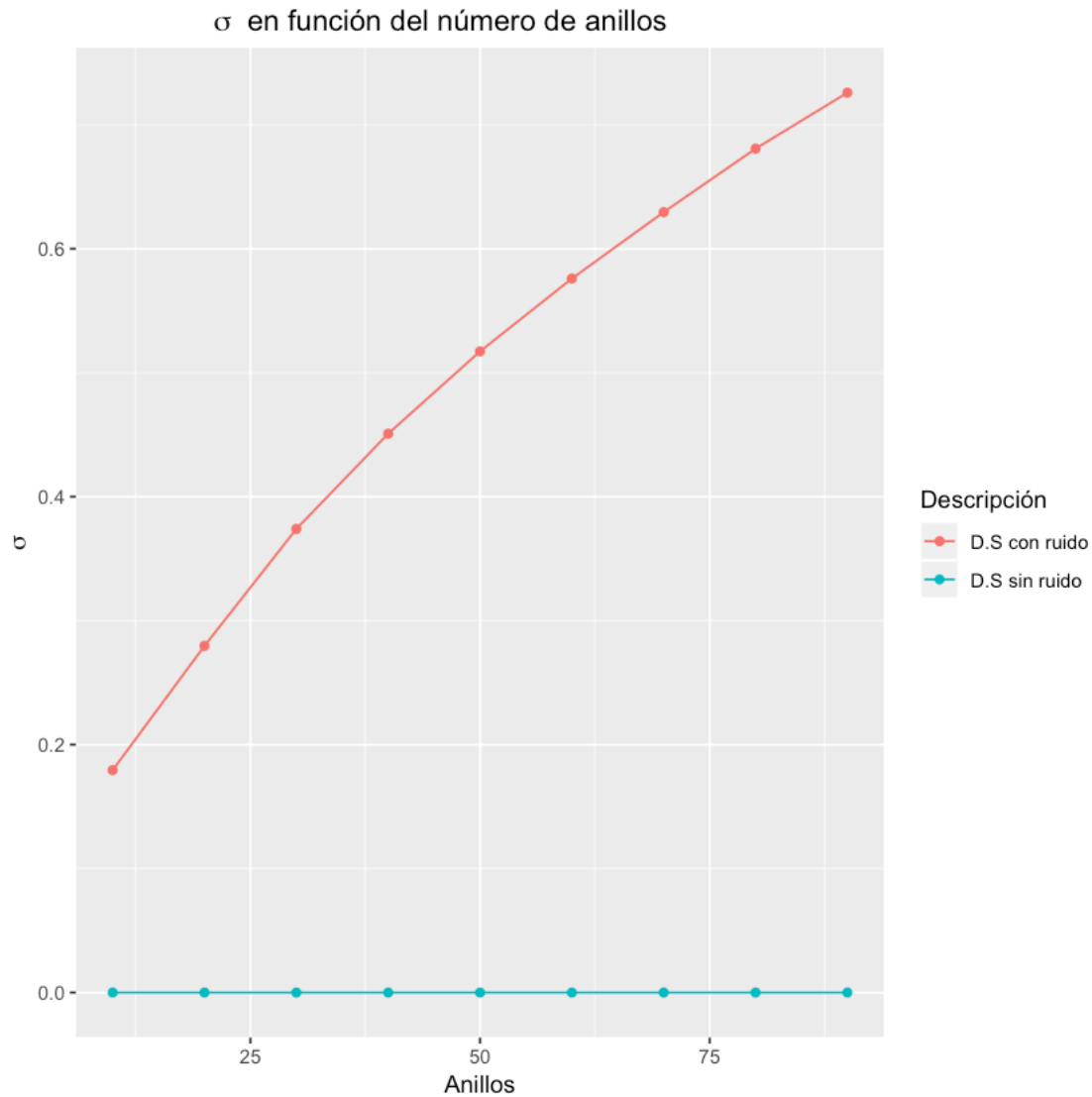


Paso seguido se verificará la desviación estandar σ en función de la tasa de ruido aplicada, fijando a 30 anillos.

Se comparan las desviaciones estandar, con ruido y sin ruido.

```
[12]: desviacion_sin_ruido = list(Descripción=c(rep("D.S sin ruido",length(ani))),
                                s.1 = as.vector(ds),
                                Anillos =ani )
desviacion_con_ruido = list(Descripción =c(rep("D.S con ruido",length(ani))),
                             s.1 =as.vector(ds_ruido),
                             Anillos = ani)

grafi_desv = bind_rows(desviacion_sin_ruido,desviacion_con_ruido)
grafi_desv
```

[13]: ##### *ds_diego_ruido_var* va en funcion de ruido, poniendo ruido del 10%
 ↳porciento así: *ds_diego_ruido_var*(30,10)

```
ds_diego_ruido_var = function(ruido){

  Anillos = 30
  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==1, {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

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

e_num_rui = matrix(0,1,Anillos)

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

##### Implementación del ruido a la funcion #####

set.seed(42)
G1_con_ruido =(sample(seq(-ruido*.01,ruido*.01,0.01),Anillos,replace = TRUE))+G1
e_num_rui = matrix(0,1,Anillos)

for (i in 1:Anillos) {
  e_num_rui[i]=(-2/((r_0/Anillos)*pi))*sum(aA_c[,i]*G1_con_ruido,na.rm = TRUE)
}
u = e_teo
e_k =e_num_rui
as.numeric(sqrt((1/Anillos)*sum((e_k-u)^2)))
}

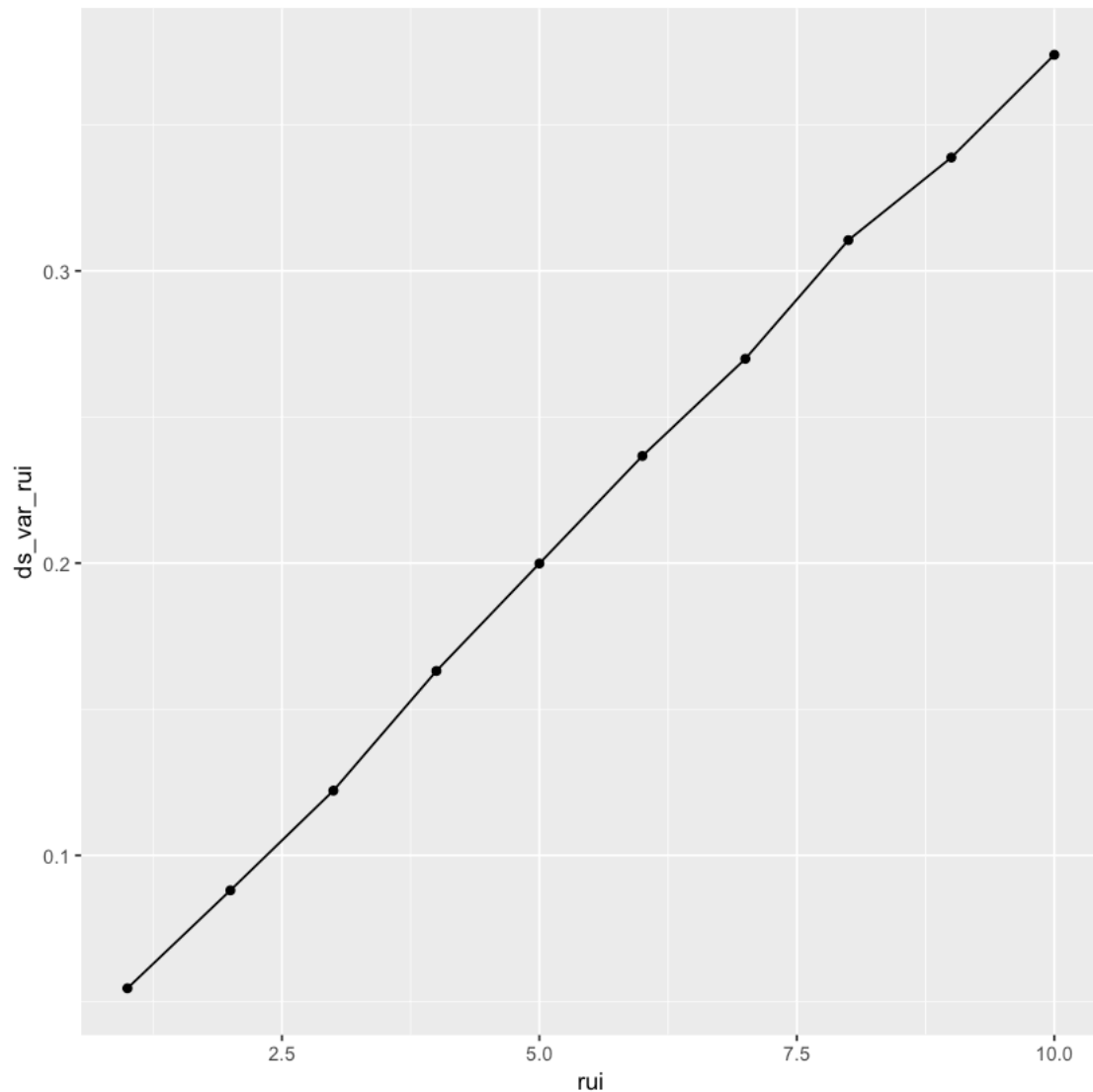
```

```
#####
#ds_ruido =sapply(ani,ds_diego_ruido)
#d_frame_ds_ruido = as.tibble(ds_ruido)%>%mutate(ani=ani)

#d_frame_ds_ruido%>%ggplot(aes(ani,ds))+
#geom_point()+
#geom_line()
```

```
[14]: ds_var_rui =sapply(seq(1,10,1),ds_diego_ruido_var)
d_frame_var_rui = as.tibble(ds_var_rui)%>%mutate(rui=seq(1,10,1))

d_frame_var_rui%>%ggplot(aes(rui,ds_var_rui))+
geom_point()+
geom_line()
```



Se procede a continuación a suavizar por medio de la función **loess** del paquete **tidyverse**, el cual hace pequeñas regresiones en ventanitas definidas (span), y crea un suavizado:

```
[15]: ds_diego_ruido_var = function(ruido){

  Anillos = 30
  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==1, {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

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

e_num_rui = matrix(0,1,Anillos)

GG1 = (1-r^2)
e_teo =(2/pi)*sqrt(1-r^2)

##### Implementación del ruido a la funcion #####

set.seed(42)
G1_con_ruido =(sample(seq(-ruido*.01,ruido*.01,0.01),Anillos,replace =
↪TRUE))+GG1
e_num_rui = matrix(0,1,Anillos)

for (i in 1:Anillos) {
  e_num_rui[i]=(-2/((r_0/Anillos)*pi))*sum(aA_c[,i]*G1_con_ruido,na.rm = TRUE)
}

G1_sin_ruido = list(metodo=c(rep("G1 sin ruido",30)),
  G = as.vector(GG1),
  radio =r )

```

```

G1_ruido = list(metodo =c(rep("G1 con ruido",30)),
                G =as.vector(G1_con_ruido),
                radio = r)

Gs = bind_rows(G1_sin_ruido,G1_ruido)

##### Tratamiento (suavisado) a los datos con ruido
↪#####
total_puntos = diff(range(G1_con_ruido))
span = 0.5/total_puntos

fit = Gs%>%filter(metodo == "G1 con ruido")%>%
  loess(G ~ radio, degree=1, span = span, data=.)

puntos_suavisados = fit$fitted

#Luego de suavizar los puntos g(r) procedemos a calcular epsilon con estos
↪valores de g(r)
for (i in 1:Anillos) {
  e_num_rui[i]=(-2/((r_0/Anillos)*pi))*sum(aA_c[,i]*puntos_suavisados,na.rm =
  ↪TRUE)
}

u = e_teo
e_k =e_num_rui
as.numeric(sqrt((1/Anillos)*sum((e_k-u)^2)))
}

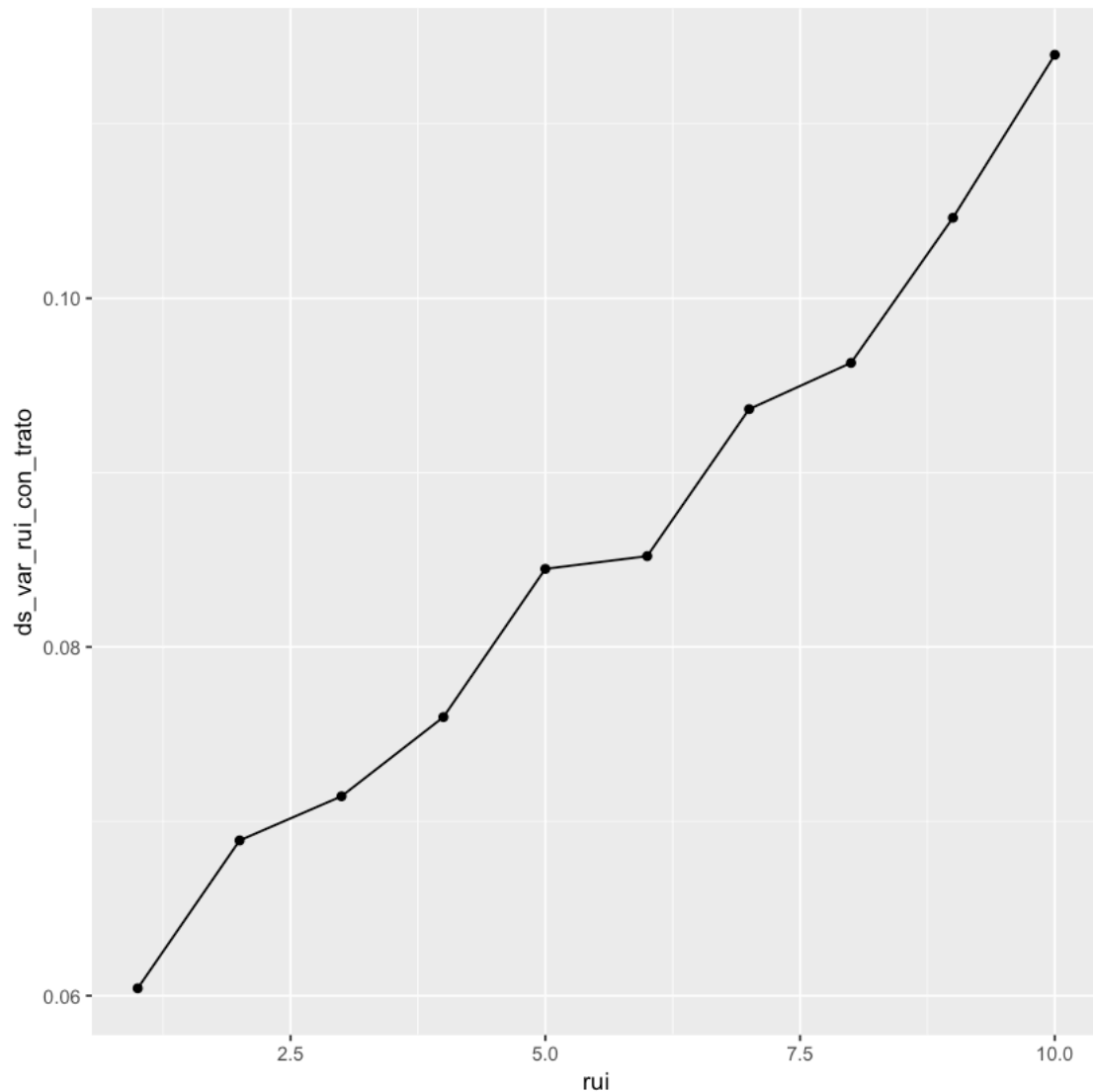
```

```

[16]: ds_var_rui_con_trato =apply(seq(1,10),ds_diego_ruido_var)
d_fra_rui_suave = as.tibble(ds_var_rui_con_trato)%>%mutate(rui=seq(1,10,1))

d_fra_rui_suave%>%ggplot(aes(rui,ds_var_rui_con_trato))+
geom_point()+
geom_line()

```



Se contrastara la desviación estandar σ cuando tenemos los datos con tratamiento de ruido, con el que tiene tratamiento.

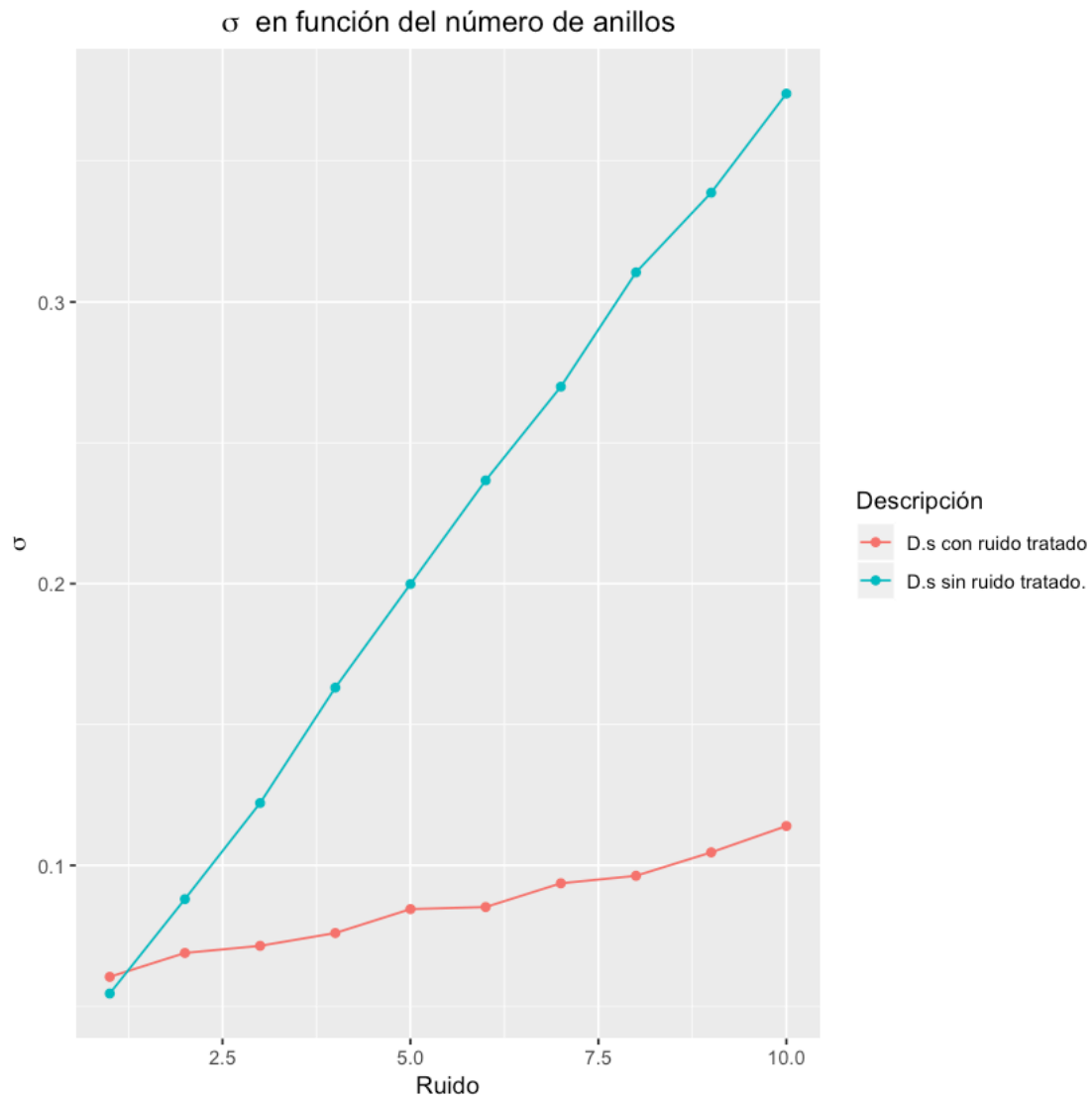
```
[17]: desvi_ruido_s_tra = list(Descripción=c(rep("D.s sin ruido tratado.",10)),
                             s.1 = as.vector(ds_var_rui),
                             Ruido =seq(1,10,1))
desviacion_con_ruido = list(Descripción =c(rep("D.s con ruido tratado",10)),
                             s.1 =as.vector(ds_var_rui_con_trato),
                             Ruido = seq(1,10,1))

grafi_desv_trat = bind_rows(desvi_ruido_s_tra,desviacion_con_ruido)

grafi_desv_trat%>%
```



```
ggplot(aes(Ruido,s.1,color = Descripción))+
  geom_line()+
  geom_point()+
  labs(title =expression("σ en función del número de
  ↪anillos"),
        y=quote(sigma),
        x=quote(Ruido))
```



```
[18]: grafi_desv_trat
save(grafi_desv_trat,file = "ds_todo_r_trata.RData")
```

Descripción	s.l	Ruido
D.s sin ruido tratado.	0.05450536	1
D.s sin ruido tratado.	0.08803000	2
D.s sin ruido tratado.	0.12214066	3
D.s sin ruido tratado.	0.16309426	4
D.s sin ruido tratado.	0.19986857	5
D.s sin ruido tratado.	0.23665665	6
D.s sin ruido tratado.	0.26992457	7
D.s sin ruido tratado.	0.31050342	8
D.s sin ruido tratado.	0.33879381	9
D.s sin ruido tratado.	0.37394718	10
D.s con ruido tratado	0.06042640	1
D.s con ruido tratado	0.06890621	2
D.s con ruido tratado	0.07143995	3
D.s con ruido tratado	0.07598251	4
D.s con ruido tratado	0.08448161	5
D.s con ruido tratado	0.08521309	6
D.s con ruido tratado	0.09364329	7
D.s con ruido tratado	0.09629883	8
D.s con ruido tratado	0.10461704	9
D.s con ruido tratado	0.11396698	10

[19]: ##### intento de quitar el ruido #####

```
G1_sin_ruido = list(Descripción=c(rep("g sin ruido",30)),
                    G = as.vector(G1),
                    radio = r )
G1_ruido = list(Descripción =c(rep("g con ruido",30)),
                G =as.vector(G1_con_ruido),
                radio = r)

Gs = bind_rows(G1_sin_ruido,G1_ruido)

total_puntos = diff(range(G1_con_ruido))
span = 0.3/total_puntos

fit = Gs%>%filter(Descripción == "g con ruido")%>%
  loess(G ~ radio, degree=1, span = span, data=.)

G1_ruido_suavizado = list(Descripción =c(rep("g con ruido tratado",30)),
                          G =as.vector(fit$fitted),
                          radio = r)
Gs = bind_rows(G1_sin_ruido,G1_ruido,G1_ruido_suavizado)

Gs%>%ggplot(aes(radio,G,col=Descripción)) +
```

```

geom_line()+
ylab("Brillo transversal g")+
xlab("Altura y normalizada")+
  theme(legend.position=c(.25, 0.3))+
  annotate(geom="text", x=.65, y=1,
↪label=TeX("$Desviación\\,estandar\\,\\sigma_1=\\,2.
↪85E-02$"),color="black",size=5)+
  annotate(geom="text", x=.75, y=0.9,
↪label=TeX("$Perfil\\,\\epsilon_1=\\sqrt{1-r^2}$"),color="black",size=5)+
  theme(#axis.text=element_text(size=13))#,
axis.title.y=element_text(size=20),
axis.title.x=element_text(size=20))

# geom_text(aes(radius,smooth,label=metodo))+
# theme(legend.position="none")+
#
puntos_suavisados = fit$fitted

u = G1
e_k =as.vector(fit$fitted)
as.numeric(sqrt((1/Anillos)*sum((e_k-u)^2)))

#labs(title=expression("          Perfil de emisividad  "*epsilon*" en
↪función del radio"),
#      x="radio de columna de plasma normalizado",
#      y=expression(" emisividad "*epsilon))+
#      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=.75, y=0.55,
↪label=TeX("$\\epsilon_1=\\frac{3200}{\\pi}\\,\\sqrt{1-r^2}$"),color="lightseagreen",size=5)

```

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"

0.0284593665144447

