

# Estimación de la precipitación en el valle de México usando datos de pluviómetros y radar meteorológico

September 15, 2021

## 1 Ejemplo de practico: estimación de la precipitación en el valle de México usando datos de pluviómetros y radar meteorológico.

Autores: Daniel Vázquez-Ramírez<sup>(1)</sup>, Martín Díaz-Viera<sup>(2)</sup>, Van Huong Le<sup>(1)</sup>

1) Posgrado en Ciencias de la Tierra

2) Instituto Mexicano del Petróleo

## 2 Introducción.

El siguiente ejemplo forma parte del artículo publicado por Diaz-Viera(2009), donde se presenta una modificación del procedimiento geoestadístico de estimación espacial de la precipitación introducido por Krajewski (1987), el cual aplica el método de cokriging ordinario, combinando imágenes de radar meteorológico con datos de pluviómetros. Aquí, a diferencia del procedimiento de Krajewski, se incluye en el método de cokriging ordinario la dependencia espacial conjunta de radar-pluviómetros mediante un modelo de correogionalización lineal. La metodología propuesta es probada usando datos de pluviógrafos y de radar de una tormenta ocurrida en el valle de la ciudad de México.

### 2.1 Abrir el proyecto en R Studio.

El proyecto precargado lo pueden descargar en la página del curso <http://www.esmg-mx.org/activities/courses/geoestadistica>. Para el análisis exploratorio descargamos la Clase Práctica 1: Análisis Exploratorio de Datos. Se descomprime el archivo y podemos ver que el proyecto se compone de un conjunto de carpetas; la carpeta "Functions" donde encontraran las funciones que usaremos durante la clase práctica y la carpeta "Scripts" donde encontrarán los archivos *Getting\_Started\_script.R* y *00Lluvia\_AED\_2D.R*. El primer script contiene las instrucciones para definir directorios, instalar los paquetes, cargar librerías y funciones, el segundo script contiene las instrucciones del ejemplo para hacer el análisis exploratorio. Les recomendamos que el archivo "lluvia.txt" lo guarden en la carpeta "Data".

Para abrir el proyecto en R Studio deben dar doble clic en *RGeoestad\_2D\_Vacio.Rproj*.

**NOTA:** es importante que el proyecto se abra usando *RGeoestad\_2D\_Vacio.Rproj*, si abren solo los scripts no podrán acceder a los directorios.

Nombre	Fecha de modificación	Tipo	Tamaño
.Rproj.user	06/09/2021 10:37 p. m.	Carpeta de archivos	
Data	06/09/2021 10:37 p. m.	Carpeta de archivos	
Documentation	19/08/2018 05:19 p. m.	Carpeta de archivos	
Functions	06/09/2021 10:37 p. m.	Carpeta de archivos	
Installation	19/08/2018 06:58 p. m.	Carpeta de archivos	
License	06/09/2021 10:37 p. m.	Carpeta de archivos	
Reports	19/08/2018 05:21 p. m.	Carpeta de archivos	
Results	10/09/2021 04:01 p. m.	Carpeta de archivos	
RGeoestad_2D_Vacio	06/09/2021 10:37 p. m.	Carpeta de archivos	
Scripts	11/09/2021 09:03 p. m.	Carpeta de archivos	
.RData	10/09/2021 06:47 p. m.	R Workspace	648 KB
.Rhistory	12/09/2021 10:03 p. m.	Archivo RHISTORY	23 KB
RGeoestad_2D_Vacio.Rproj	12/09/2021 09:42 p. m.	R Project	1 KB

## 2.2 Carga de paqueterías y funciones.

Para obtener la estimación espacial debemos instalar en R Studio los siguientes paquetes: Rcpp, maps, mapproj, actuar, fields, fitdistrplus, geoR, gstat, MASS, moments, powerLaw, RFOC, spatstat, ADGofTest, reshape, sp.

**NOTA: la versión de R recomendada es 4.1.1**, de lo contrario, no podrán instalar el paquete “actuar”.

Hay dos formas de instalar estos paquetes: la primera opción es ir a la barra de menús en la interfaz de R Studio, dar click en *tools > installPackages*. En el renglón Packages pondrán los nombres de los paquetes separados por coma y después dan click en install.

La segunda opción es usando la consola, para eso abrimos el script *Getting\_Started\_script.R* y ejecutar las siguientes líneas.

```
[1]: root_dir<-getwd()

#install_dir- installation directory

install_dir<-paste(root_dir,"/Installation",sep="")

setwd(install_dir)

install.packages("Rcpp")
install.packages("maps")
install.packages("mapproj")
install.packages("actuar")
install.packages("fields")
install.packages("fitdistrplus")
install.packages("geoR")
install.packages("gstat")
install.packages("MASS")
install.packages("moments")
```

```
install.packages("powerLaw")
install.packages("RFOC")
install.packages("spatstat")
install.packages("ADGofTest")
install.packages("reshape")
install.packages("sp")

#set back to root work directory
```

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'Rcpp' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'maps' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'mapproj' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'actuar' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'fields' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'fitdistrplus' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'geoR' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'gstat' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'MASS' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'moments' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'powerLaw' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'RFOC' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'spatstat' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'ADGofTest' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'reshape' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Installing package into 'E:/OneDrive/Documentos/R/win-library/4.1'  
(as 'lib' is unspecified)

package 'sp' successfully unpacked and MD5 sums checked

The downloaded binary packages are in  
C:\Users\danie\AppData\Local\Temp\Rtmp0YmHlY\downloaded\_packages

Despues de instalar los paquetes debemos cargarlos de la siguiente forma:

```
[2]: setwd(root_dir)

#### Load Packages ####
```

```
library(actuar)
library(Rcpp)
library(maps)
library(mapproj)
library(fields)
library(fitdistrplus)
library(geoR)
library(gstat)
library(MASS)
library(moments)
library(powerLaw)
library(RFOC)
library(spatstat)
library(ADGofTest)
library(reshape)
library(sp)
```

Attaching package: 'actuar'

The following object is masked from 'package:grDevices':

cm

Loading required package: spam

Loading required package: dotCall64

Loading required package: grid

Spam version 2.7-0 (2021-06-25) is loaded.

Type 'help( Spam)' or 'demo( spam)' for a short introduction and overview of this package.

Help for individual functions is also obtained by adding the suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

Attaching package: 'spam'

The following objects are masked from 'package:base':

backsolve, forwardsolve

Loading required package: viridis

Loading required package: viridisLite

Attaching package: 'viridis'

The following object is masked from 'package:maps':

unemp

See <https://github.com/NCAR/Fields> for  
an extensive vignette, other supplements and source code

Loading required package: MASS

Loading required package: survival

```
-----  
Analysis of Geostatistical Data  
For an Introduction to geoR go to http://www.leg.ufpr.br/geoR  
geoR version 1.8-1 (built on 2020-02-08) is now loaded  
-----
```

Loading required package: spatstat.data

Loading required package: spatstat.geom

spatstat.geom 2.2-2

Attaching package: 'spatstat.geom'

The following object is masked from 'package:MASS':

area

The following object is masked from 'package:actuar':

discretise

Loading required package: spatstat.core

Loading required package: nlme

Loading required package: rpart

spatstat.core 2.3-0

Attaching package: 'spatstat.core'

The following object is masked from 'package:gstat':

idw

Loading required package: spatstat.linnet

spatstat.linnet 2.3-0

spatstat 2.2-0 (nickname: 'That's not important right now')

For an introduction to spatstat, type 'beginner'

Comprobamos que todos los paquetes hayan sido cargados, si es así, cargaremos las funciones. Estas nos permitirán obtener los gráficos, modelos, etc.

```
[3]: root_dir<-getwd()

function_dir<-paste(root_dir,"/Functions",sep="")

setwd(function_dir)

source("AllModel.R", encoding='ISO-8859-1')
source("BasicStats.R", encoding='ISO-8859-1')
source("BestModel.R", encoding='ISO-8859-1')
source("BestModel.R")
source("BestModelName.R")
source("CDF.R")
source("CoKrigingOrd.R")
source("CoKrigingOrdAnis.R")
source("CrossValidation.R")
source("CrossValidation2.R")
source("CrossVariograma.R")
source("DEspacial.R", encoding='ISO-8859-1')
source("Distance.R")
source("Estadisticas.R")
```



```

source("EyeModel.R", encoding='ISO-8859-1')
source("FitDistribution.R", encoding='ISO-8859-1')
source("GDEspacial.R", encoding='ISO-8859-1')
source("GDirecciones.R", encoding='ISO-8859-1')
source("GNormal.R", encoding='ISO-8859-1')
source("hist2.R")
source("HistBoxplot.R")
source("HistModel.R")
source("KrigingOrd.R", encoding='ISO-8859-1')
source("KrigingOrdAnis.R", encoding='ISO-8859-1')
source("ModelVariogram.R")
source("Modelo.R")
source("Outliers.R")
source("OutliersCount.R")
source("OutliersCountTwo.R")
source("OutliersPos.R")
source("OutliersTwo.R")
source("PPplot.R")
source("QQplot.R")
source("RangoParams.R")
source("Regresion.R")
source("ScatterPlot.R")
source("Tendencia.R")
source("Transformacion.R")
source("Trend.R")
source("Val_Estadisticos.R", encoding='ISO-8859-1')
source("Validacion.R", encoding='ISO-8859-1')
source("ValidacionCross.R", encoding='ISO-8859-1')
source("Variograma.R")
source("Variograma4D.R", encoding='ISO-8859-1')
source("hist2.R")
source("scaterplot.R")
source("scaterplotReg.R")
#source("PlotGridCells.R")

setwd(root_dir)

```

Para saber si todos los directorios fueron cargados adecuadamente, veremos que el directorio final mostrado en la consola es "RGeoestad\_2D\_Vacio" como se muestra en la siguiente imagen.

De lo contrario, se debe poner en la consola la instrucción "setwd(root\_dir)" o bien se puede redirigir el directorio en la barra de menus>Session>Set Working Directory>To Project Directory.

Despues de cargar las funciones podemos continuar con el análisis exploratorio.

Es importante recordar que cada vez que se abre el proyecto en R Studio se deben cargar solo las librerias como se muestra en la siguiente imagen.

**NOTA:** Si este proceso se hace en este notebook, se deben cargar solo las librerias y las funciones.

## 2.3 Carga de datos.

Ahora tenemos que cargar los datos de cada variable con su respectiva posición espacial en coordenadas UTM. Es importante que cada columna tenga su propio encabezado, así será fácil localizarlos e indexarlos. Para seleccionar el archivo que contiene la información que necesitamos, ejecutamos el comando “read.table”, el cual contiene las siguientes tres instrucciones: file=file.choose(), esta instrucción indica que quieres seleccionar el archivo usando una ventana emergente similar a la mostrada en la siguiente imagen:

header=TRUE indica que las columnas tienen encabezado y na.strings=”-999.25” es una condicional para que cualquier celda nula sea llenada con el número -999.25.

```
[6]: Data_File <- read.table(file=file.choose(),header=TRUE,na.strings="-999.25")
```

Si se desea abrir un archivo con información delimitada por comas (.csv) se cambia la instrucción por: “Data\_File <- read.csv(file=file.choose(),header=T,na.strings=”-999.25”)”

Para ordenar los resultados necesitamos crear una carpeta que usemos específicamente para el análisis exploratorio de datos (AED), ahí se almacenarán tablas e imágenes, esto lo hacemos con el comando “dir.create”, donde le indicaremos la ruta donde se creará la carpeta AED.

**Nota: no es necesario ejecutar esta línea más de una vez, de lo contrario R Studio mostrará un error “Warning message in dir.create(paste(getwd(),”/Results/AED”, sep = “”) already exists”**

```
[7]: dir.create(paste(getwd(),"/Results/AED", sep=""))
```

```
Warning message in dir.create(paste(getwd(), "/Results/AED", sep = "")):
"'C:\Users\danie\Dropbox\semestre 2022-1\clase geoestadistica\ejemplo
lluvia\Results\AED' already exists"
```

## 3 Análisis exploratorio de datos.

Como se mostró en clase, el objetivo del análisis exploratorio es examinar las variables aleatorias disponibles y establecer si estas cumplen con los supuestos que requiere la estimación. Por lo tanto, debemos verificar su normalidad, linealidad, homocedasticidad, identificar los valores atípicos (outliers) y evaluar el impacto que tendrán estos valores durante el análisis variográfico y por supuesto, la estimación.

Para este ejemplo las variables son los valores obtenidos de los pluviómetros (Pluv\_mm) y del radar meteorológico (Radar\_mm), los cuales tienen una distribución espacial en coordenadas UTM.

Después de cargar el archivo con la información y asignarle el nombre “Data\_File”, necesitamos las variables aleatorias y su posición espacial. Esto lo podemos hacer de la siguiente forma:

```
[8]: XCoord<-Data_File$UTM_X_m      #Coordenada UTM en x
YCoord<-Data_File$UTM_Y_m      #Coordenada UTM en y
Radar_mm<-Data_File$Radar_mm    #variable con la información del radar
    ↪ meteorológico
Pluv_mm<-Data_File$Pluv_mm      #variable con la información de los pluviómetros
```

Ya que tenemos las variables necesitamos saber sobre sus estadígrafos, esto lo podemos calcular usando la función “Estadísticas”. Es importante mencionar que los valores calculados en este paso se usarán en los gráficos.

```
[9]: XCoord_Stat<-Estadisticas(XCoord)
      YCoord_Stat<-Estadisticas(YCoord)
      Radar_mm_Stat<-Estadisticas(Radar_mm)
      Pluv_mm_Stat<-Estadisticas(Pluv_mm)
```

### 3.1 Análisis estadístico univariado.

Para la interpretación estadística univariada comenzaremos dos elementos: la tabla con los valores estadísticos y el histograma con boxplot. La tabla con los estadígrafos la obtenemos usando la función “Val\_Estadísticos” y la guardamos en la carpeta “AED” con la instrucción “write.csv”.

```
[10]: Data_File_Stat <- Val_Estadísticos(Data_File)
      write.csv(Data_File_Stat , file = "Results/AED/Data_File_Stat.csv") #esta linea
      →sirve para guardar los resultados en un archivo csv
      print(Data_File_Stat[,3:4])
```

	Radar_mm	Pluv_mm
No_muestras	50.00000	50.00000
Minimo	0.18000	0.25000
Cuartil_1er	0.87750	0.31250
Mediana	1.42500	1.00000
Media	1.83500	1.47000
Cuartil_3er	2.37500	1.93750
Maximo	7.79000	7.75000
Rango	7.61000	7.50000
Rango_Intercuartil	1.49750	1.62500
Varianza	2.47352	2.57561
Desv_Estandar	1.57274	1.60487
Simetria	2.02492	2.46929
Curtosis	7.99412	10.05416

#### 3.1.1 Análisis estadístico univariado para el radar metereológico (Radar\_mm).

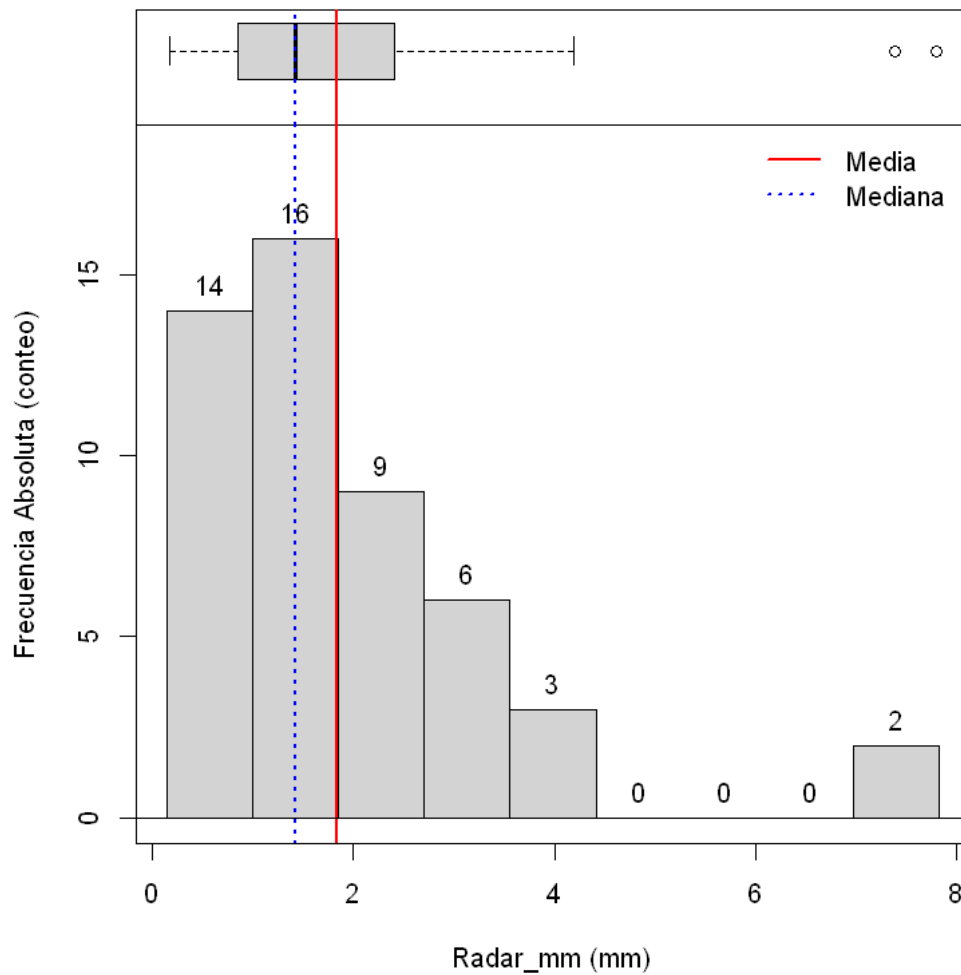
El histograma con boxplot se grafica usando la función “HistBoxplot”, esta necesita que le indiquemos el vector con los valores (x), su valor esperado (mean) y su mediana (median). Si observan la instrucción para obtener el histograma para el radar meteorológico, el valor esperado y la mediana la toma de la tabla que se generó con la función “Estadísticas”. Por ejemplo, la instrucción “Radar\_mm\_Stat[5,2]” indica que el valor esperado se encuentra en la fila 5 columna 2. también podemos indicar el número de bins que necesitamos, en este caso es “nbin = 9”, es importante recordar que el número por default que encontrarán es 9, pero puede cambiarse según las necesidades del usuario.

La función “HistBoxplot” puede generar dos tipos de histograma: de frecuencia absoluta y de frecuencia relativa. Para obtener el histograma de frecuencia absoluta debemos indicar que “AbsFreq = TRUE, PercentFreq = FALSE” y para el histograma de frecuencia relativa debemos indicar

que “AbsFreq = FALSE, PercentFreq = TRUE”.

El histograma para el radar meteorológico (Radar\_mm) con frecuencia absoluta es:

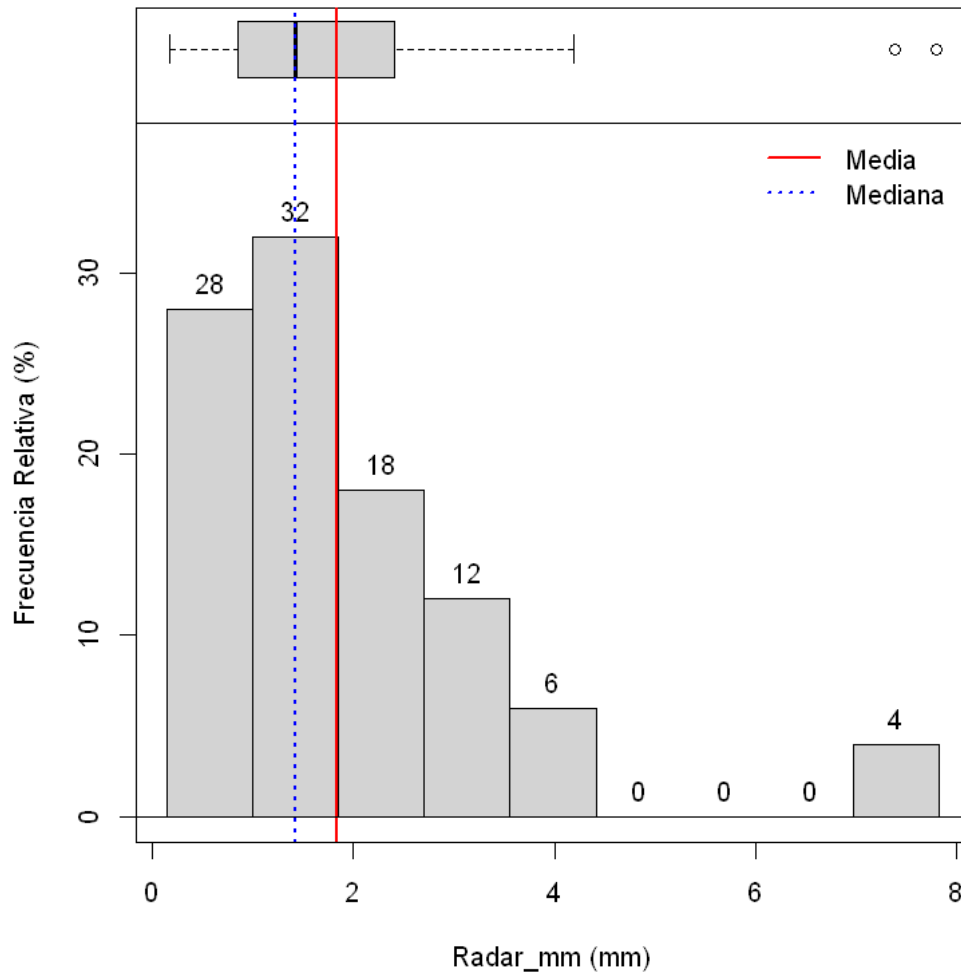
```
[11]: HistBoxplot(x=Radar_mm, mean = Radar_mm_Stat[5,2], median = Radar_mm_Stat[4,2],  
  →main = "",  
        xlab = "Radar_mm (mm)", ylab = "Frecuencia Absoluta (conteo)",  
  →AbsFreq = TRUE, PercentFreq = FALSE,  
        nbin = 9)
```



Y el histograma del radar meteorológico con frecuencia relativa es:

```
[10]: HistBoxplot(x=Radar_mm, mean = Radar_mm_Stat[5,2], median = Radar_mm_Stat[4,2],  
  →main = "",
```

```
xlab = "Radar_mm (mm)", ylab = "Frecuencia Relativa (%)", AbsFreq =   
→FALSE, PercentFreq = TRUE,   
nbin = 9)
```



Analizando los histogramas y los estadígrafos de la variable del radar meteorológico (Radar\_mm) tiene una diferencia entre la media y la mediana de 0.47, su coeficiente de asimetría es de 2.46929, lo cual significa que la variable es asimétrica. Esto se confirma con los histogramas, los cuales muestran que la asimetría es positiva. También podemos notar que el boxplot muestra dos valores atípicos localizados en el extremo derecho. El valor de la curtosis es de 7.99412, lo cual nos indica que es leptocúrtica.

Para saber cuáles son esos valores atípicos usamos la función "OutliersPos".

```
[12]: Radar_mm_outliers<-OutliersPos(Radar_mm)
Data_File[Radar_mm_outliers,c(1,2,3)]
```

A data.frame: 2 × 3		UTM_X_m	UTM_Y_m	Radar_mm
		<int>	<int>	<dbl>
	32	482135	2129478	7.79
	36	497898	2129469	7.39

Ahí podemos notar que la muestra 32 y 36 son los valores atípicos, por lo que retiramos esos valores de la variable, esto lo hacemos usando una variable nueva que llamaremos “Radar\_mm\_out”. También necesitamos calcular los valores estadísticos de esta nueva variable.

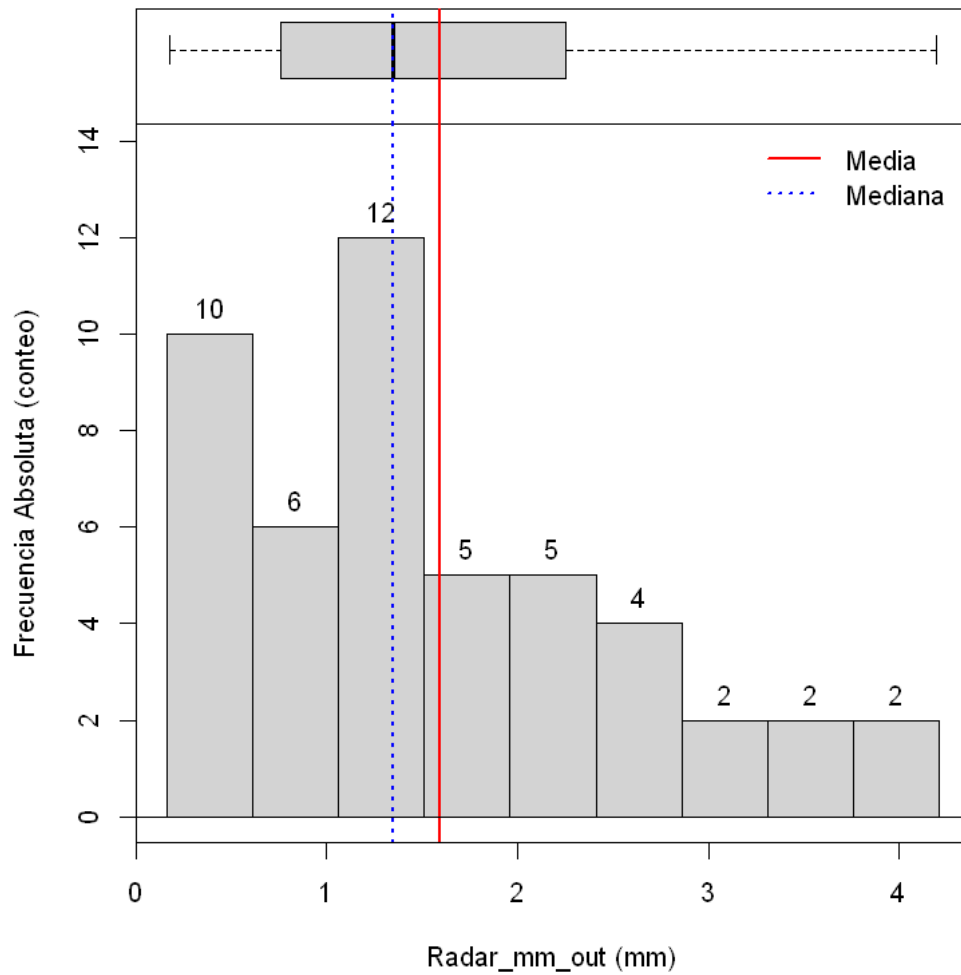
**NOTA: retirar los valores atípicos no significa que no usaremos más adelante, los necesitaremos para saber si** Por ejemplo, el vector “Radar\_mm” es el vector original, el vector “Radar\_mm\_out” contiene todos los valores excepto aquellos que fueron señalados como valores atípicos.

```
[13]: Radar_mm_out<-Radar_mm[-Radar_mm_outliers]
Radar_mm_out_Stat<-Estadisticas(Radar_mm_out)
Radar_mm_out_Stat
```

A data.frame: 14 × 2		Statistics	Values
		<chr>	<dbl>
	muestras	n	48.0000
	minimos	Minimum	0.1800
	cuantiles1	1st. Quartile	0.8050
	medianas	Median	1.3500
	medias	Mean	1.5952
	cuantiles3	3rd. Quartile	2.2475
	maximos	Maximum	4.1900
	rangos	Rank	4.0100
	rangosInt	Interquartile Rank	1.4425
	varianzas	Variance	1.1090
	desvs	Standard Deviation	1.0531
	CVs	Variation Coeff.	0.6602
	simetrias	Skewness	0.6937
	curtosiss	Kurtosis	2.7723

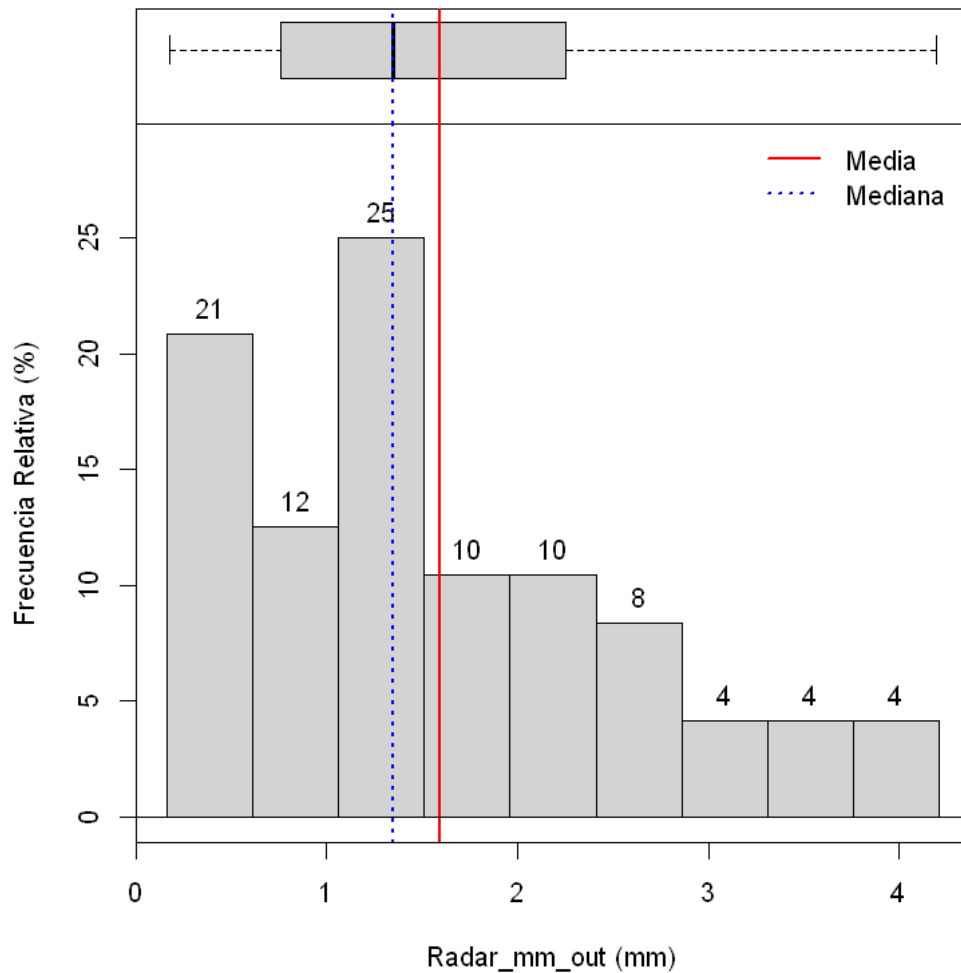
Volvemos a graficar el histograma para los valores obtenidos con el radar metereológico sin valores atípicos (Radar\_mm\_out) con frecuencia absoluta.

```
[14]: HistBoxplot(x=Radar_mm_out, mean = Radar_mm_out_Stat[5,2], median =
→Radar_mm_out_Stat[4,2], main = "",
xlab = "Radar_mm_out (mm)", ylab = "Frecuencia Absoluta (conteo)",
→AbsFreq = TRUE, PercentFreq = FALSE,
nbin = 9)
```



Y el histograma para los valores obtenidos con el radar metereológico sin valores atípicos (Radar\_mm\_out) con frecuencia relativa.

```
[15]: HistBoxplot(x=Radar_mm_out, mean = Radar_mm_out_Stat[5,2], median = Radar_mm_out_Stat[4,2], main = "",
→          xlab = "Radar_mm_out (mm)", ylab = "Frecuencia Relativa (%)",
→          AbsFreq = FALSE, PercentFreq = TRUE,
→          nbin = 9)
```



Analizando la diferencia entre media y la mediana de esta nueva variable es de 0.2452 y su asimetría es de 0.6937. En principio esto nos indica que la asimetría disminuyó, sin embargo, la variable sigue mostrando asimetría positiva. Con el caso de la curtosis, el valor pasó de 7.99412 a 2.7723, con lo cual consideramos que es planicúrtica.

Ahora podemos observar que los boxplot no muestran nuevos valores atípicos para la variable del radar meteorológico sin valores atípicos (Radar\_mm\_out). Esto lo podemos confirmar usando la función “OutliersPos”.

```
[16]: Radar_mm_out_outliers<-OutliersPos(Radar_mm_out)
      print(Radar_mm_out_outliers)
```

```
numeric(0)
```



**Transformación de variable para el radar meteorológico (Radar\_mm).** Dado que no se logró obtener la normalidad en esta variable, podemos usar alguna transformación.

En estadística, la transformación de datos es la aplicación de una función matemática determinada a cada punto en un conjunto de datos, es decir, cada punto de datos  $z_i$  se reemplaza con el valor transformado  $y_i = f(z_i)$ , donde  $f$  es una función.

Las transformaciones generalmente se aplican para que los datos parezcan cumplir más con los supuestos de un procedimiento de inferencia estadística que se aplicará o para mejorar la interpretabilidad o la apariencia de los gráficos.

Las razones más comunes para aplicar una transformación son:

- Reducir la asimetría.
- Lograr relaciones de dependencia lineales o cuasi lineales
- Conveniencia.

Las transformaciones más comunes son:

Asimetrías positivas	Ecuación	Asimetrías negativas	Ecuación
Raíz cuadrada	$v_{at} = \sqrt{v_a}$	Potencias	$v_{at} = v_a^n$
Logarítmica	$v_{at} = \text{Log}(v_a)$	Arcseno	$v_{at} = \text{arcsen}(v_a)$
Recíproca	$v_{at} = \frac{1}{v_a}$	Exponencial	$v_{at} = \exp(v_a)$

Donde  $v_a$  es la variable aleatoria y  $v_{at}$  es la variable aleatoria transformada.

**NOTA: El usuario debe saber si la transformación puede ser usada en la variable aleatoria.**

Por ejemplo, si la variable aleatoria contiene muestras con valor cero, entonces no podrá usar las transformaciones logarítmica o recíproca. Si desea usar la transformación arcoseno, la variable aleatoria debe tener componente trigonométrica.

Dado que la variable del radar meteorológico (Radar\_mm) presenta asimetría positiva, se van a usar dos tipos de transformaciones: raíz cuadrada y logarítmica.

### Transformación de raíz cuadrada

Empezaremos con la transformación de raíz cuadrada usando el vector "Radar\_mm", la cual se hace de la siguiente forma:

```
[19]: Data_File$Radar_mm_Sqrt<-sqrt(Radar_mm)
      Radar_mm_Sqrt <- Data_File$Radar_mm_Sqrt
```

Ya que obtenemos la variable aleatoria transformada por raíz cuadrada (Radar\_mm\_Sqrt), debemos obtener sus estadígrafos.

```
[20]: # Estadística básica
      Radar_mm_Sqrt_Stat<-Estadisticas(Radar_mm_Sqrt) #estos valores los necesitamos
      ↪para poder graficar los histogramas

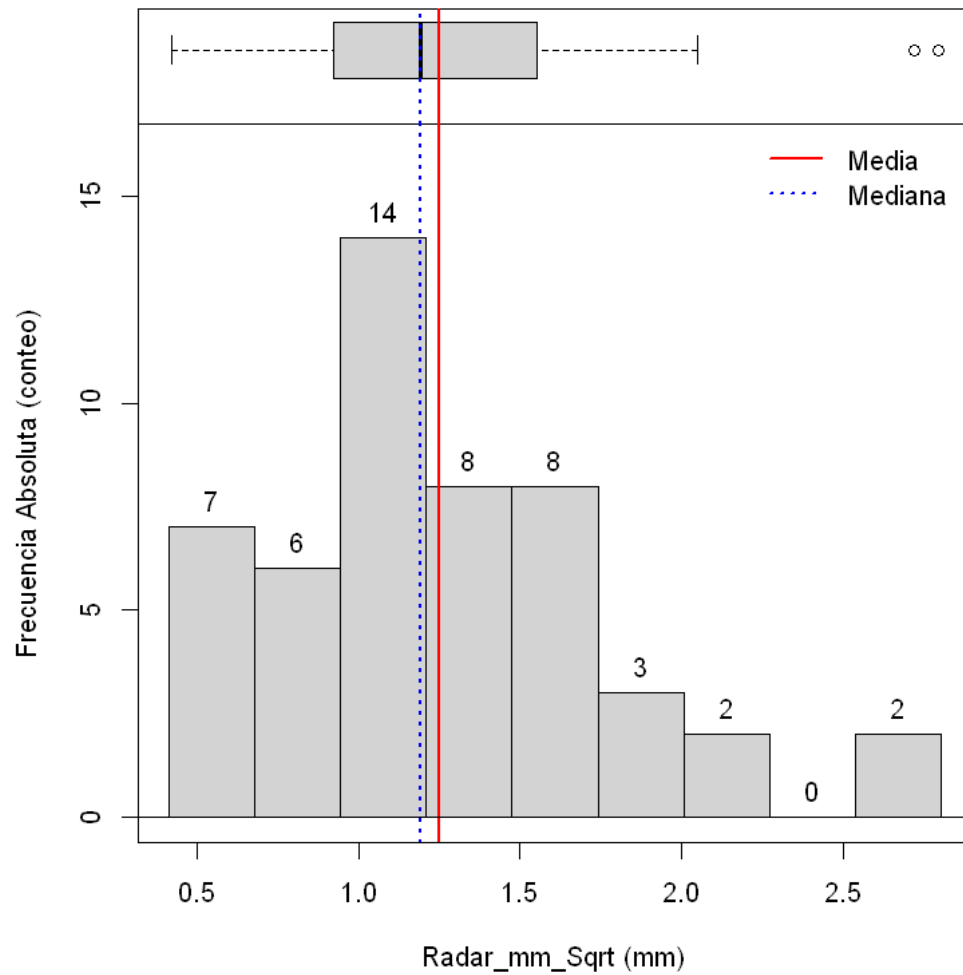
      # Estadística básica comparativa entre (Radar_mm) y la transformación
      ↪(Radar_mm_Sqrt)
      Data_File_Stat <- Val_Estadisticos(Data_File)
```

```
Data_File_Stat[,c(3,5)]
```

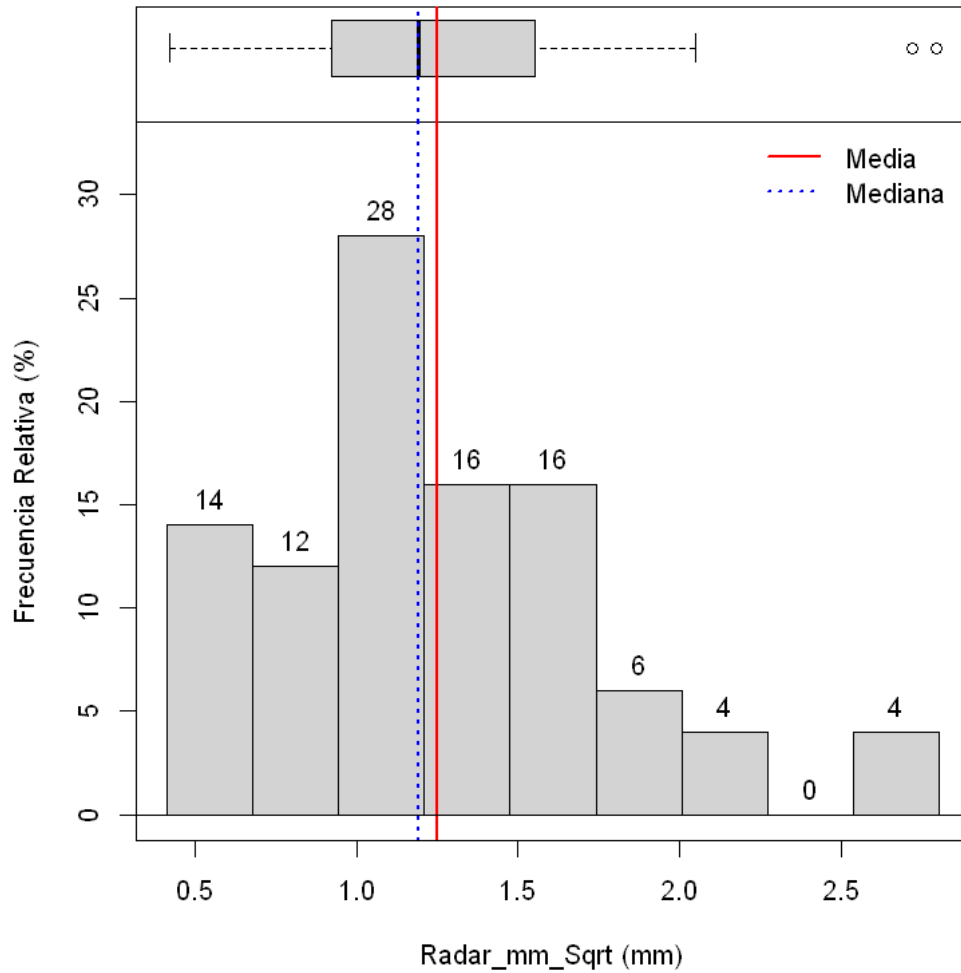
	Radar_mm <dbl>	Radar_mm_Sqrt <dbl>
No_muestras	50.00000	50.00000
Minimo	0.18000	0.42426
Cuartil_1er	0.87750	0.93641
Mediana	1.42500	1.19364
Media	1.83500	1.25106
Cuartil_3er	2.37500	1.54098
Maximo	7.79000	2.79106
Rango	7.61000	2.36679
Rango_Intercuartil	1.49750	0.60456
Varianza	2.47352	0.27536
Desv_Estandar	1.57274	0.52474
Simetria	2.02492	0.74939
Curtosis	7.99412	3.86947

Y graficamos los histogramas con frecuencia absoluta y relativa de la transformación raíz cuadrada.

```
[21]: HistBoxplot(x=Radar_mm_Sqrt, mean = Radar_mm_Sqrt_Stat[5,2], median =  
  ↳Radar_mm_Sqrt_Stat[4,2], main = "",  
    xlab = "Radar_mm_Sqrt (mm)", ylab = "Frecuencia Absoluta (conteo)",  
  ↳AbsFreq = TRUE, PercentFreq = FALSE,  
    nbin = 9)
```



```
[22]: HistBoxplot(x=Radar_mm_Sqrt, mean = Radar_mm_Sqrt_Stat[5,2], median = Radar_mm_Sqrt_Stat[4,2], main = "",
  ↳ xlab = "Radar_mm_Sqrt (mm)", ylab = "Frecuencia Relativa (%)",
  ↳ AbsFreq = FALSE, PercentFreq = TRUE,
  ↳ nbin = 9)
```



Si observamos la diferencia entre la media y la mediana de la variable transformada (Radar\_mm\_Sqrt) podemos notar que es de 0.05742, lo cual podemos considerar como muy bajo en comparación de la diferencia obtenida de la variable sin transformar que es de 0.47, sin embargo, observamos que el boxplot muestra la presencia de dos valores atípicos, los cuales debemos retirar y verificar que no afecten la supuesta simetría que hemos logrado con la transformación.

```
[23]: Radar_mm_Sqrt_outliers<-OutliersPos(Radar_mm_Sqrt)
      Data_File[Radar_mm_Sqrt_outliers,c(1,2,5)]
```

A data.frame: 2 × 3		UTM_X_m	UTM_Y_m	Radar_mm_Sqrt
		<int>	<int>	<dbl>
	32	482135	2129478	2.791057
	36	497898	2129469	2.718455

Retiramos los valores atípicos y obtenemos los valores estadísticos.

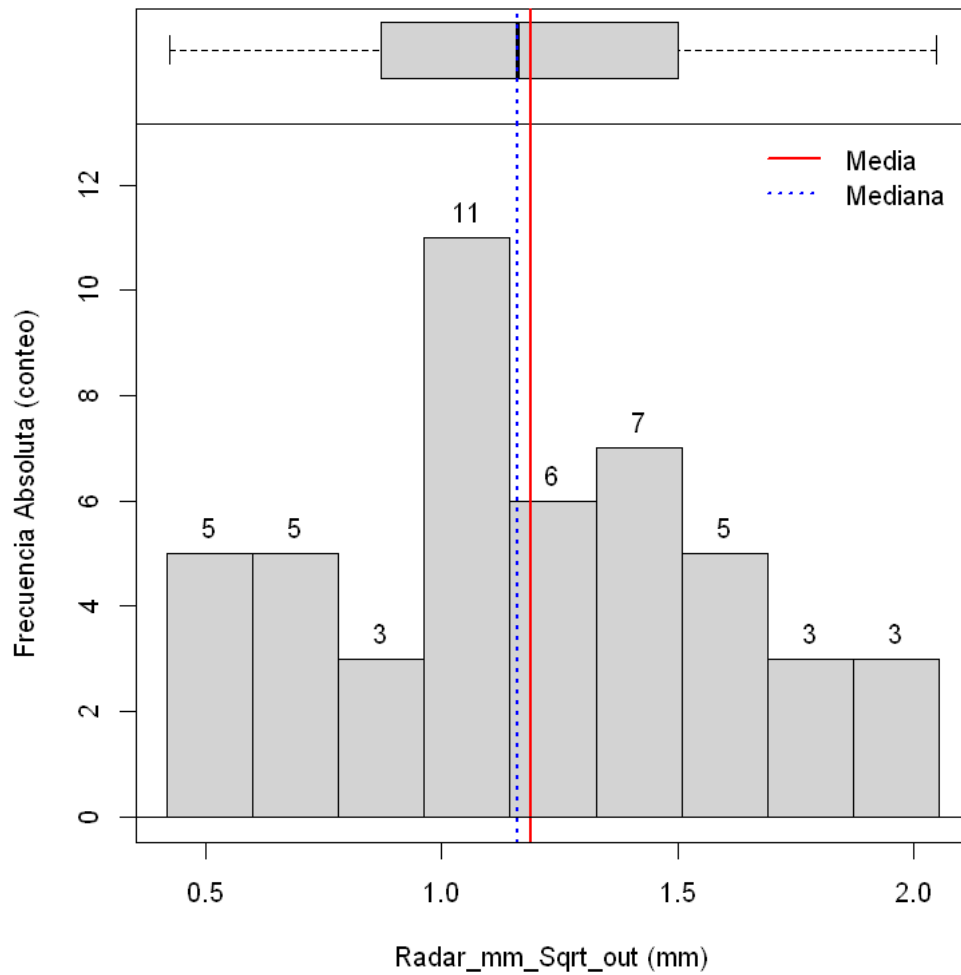
```
[24] : Radar_mm_Sqrt_out<-Radar_mm_Sqrt[-Radar_mm_Sqrt_outliers]
Radar_mm_Sqrt_out_stat<-Estadisticas(Radar_mm_Sqrt_out)
Radar_mm_Sqrt_out_stat
```

A data.frame: 14 × 2

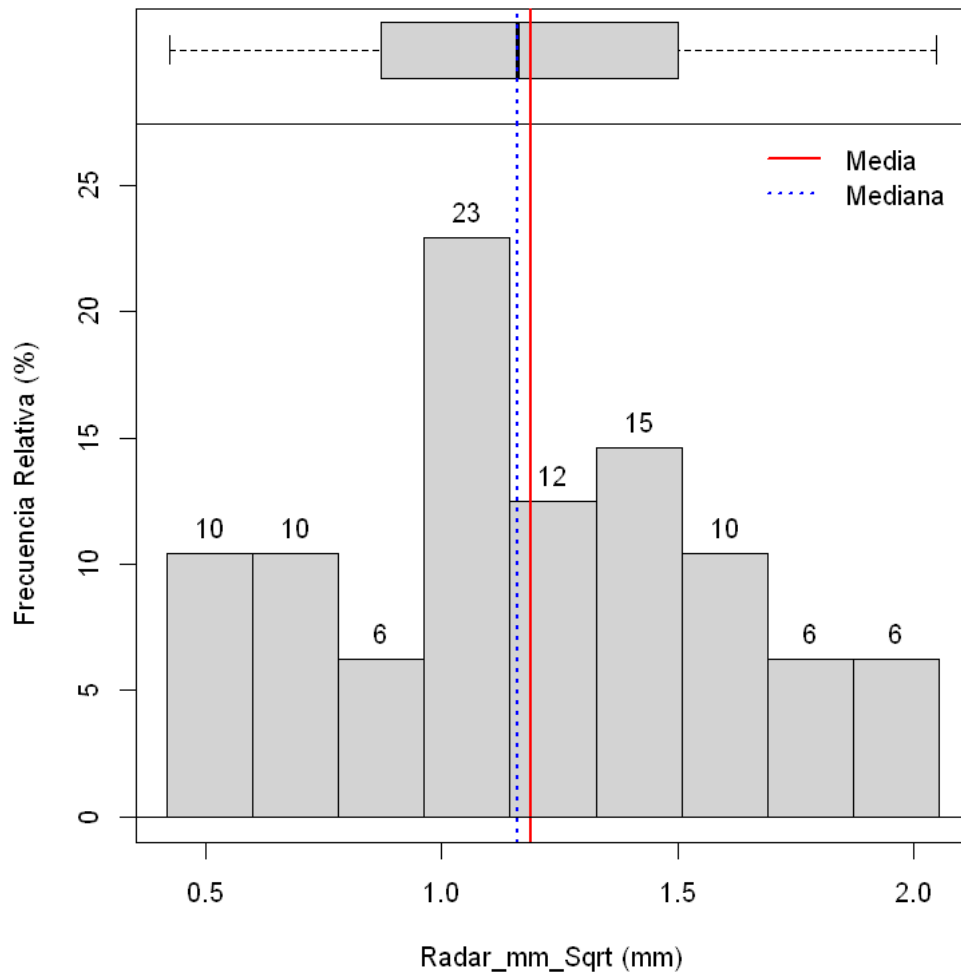
	Statistics <chr>	Values <dbl>
muestras	n	48.0000
minimos	Minimum	0.4243
cuantiles1	1st. Quartile	0.8961
medianas	Median	1.1618
medias	Mean	1.1884
cuantiles3	3rd. Quartile	1.4992
maximos	Maximum	2.0469
rangos	Rank	1.6227
rangosInt	Interquartile Rank	0.6031
varianzas	Variance	0.1868
desvs	Standard Deviation	0.4322
CVs	Variation Coeff.	0.3637
simetrias	Skewness	0.0273
curtosiss	Kurtosis	2.2704

Y graficamos sus histogramas.

```
[25] : HistBoxplot(x=Radar_mm_Sqrt_out, mean = Radar_mm_Sqrt_out_stat[5,2], median =
→Radar_mm_Sqrt_out_stat[4,2], main = "",
      xlab = "Radar_mm_Sqrt_out (mm)", ylab = "Frecuencia Absoluta",
→(conteo)", AbsFreq = TRUE, PercentFreq = FALSE,
      nbin = 9)
```



```
[26]: HistBoxplot(x=Radar_mm_Sqrt_out, mean = Radar_mm_Sqrt_out_stat[5,2], median =
→Radar_mm_Sqrt_out_stat[4,2], main = "",
        xlab = "Radar_mm_Sqrt (mm)", ylab = "Frecuencia Relativa (%)",
→AbsFreq = FALSE, PercentFreq = TRUE,
        nbin = 9)
```



Confirmamos que no existan más valores atípicos.

[27]: `OutliersPos(Radar_mm_Sqrt_out)`

Con los resultados obtenidos al retirar los dos valores atípicos podemos notar que la diferencia entre la media y la mediana pasaron de 0.05742 a 0.0266, podemos considerar que la transformación cumplió su propósito al reducir de manera significativa la asimetría, sin embargo, comparando las muestras transformadas con y sin valores atípicos podría no justificar el retiro de los valores atípicos ya que es muy pequeña la diferencia.

### Transformación logarítmica

Ahora probaremos usando la transformación logarítmica. La cual se hace de la siguiente forma:

```
[28]: Data_File$Radar_mm_Log<-log(Radar_mm)
Radar_mm_Log <- Data_File$Radar_mm_Log
```

y al igual que la transformada por raíz cuadrada, obtenemos sus valores estadísticos.

```
[29]: # Basic Statistics
Radar_mm_Log_Stat<-Estadisticas(Radar_mm_Log)

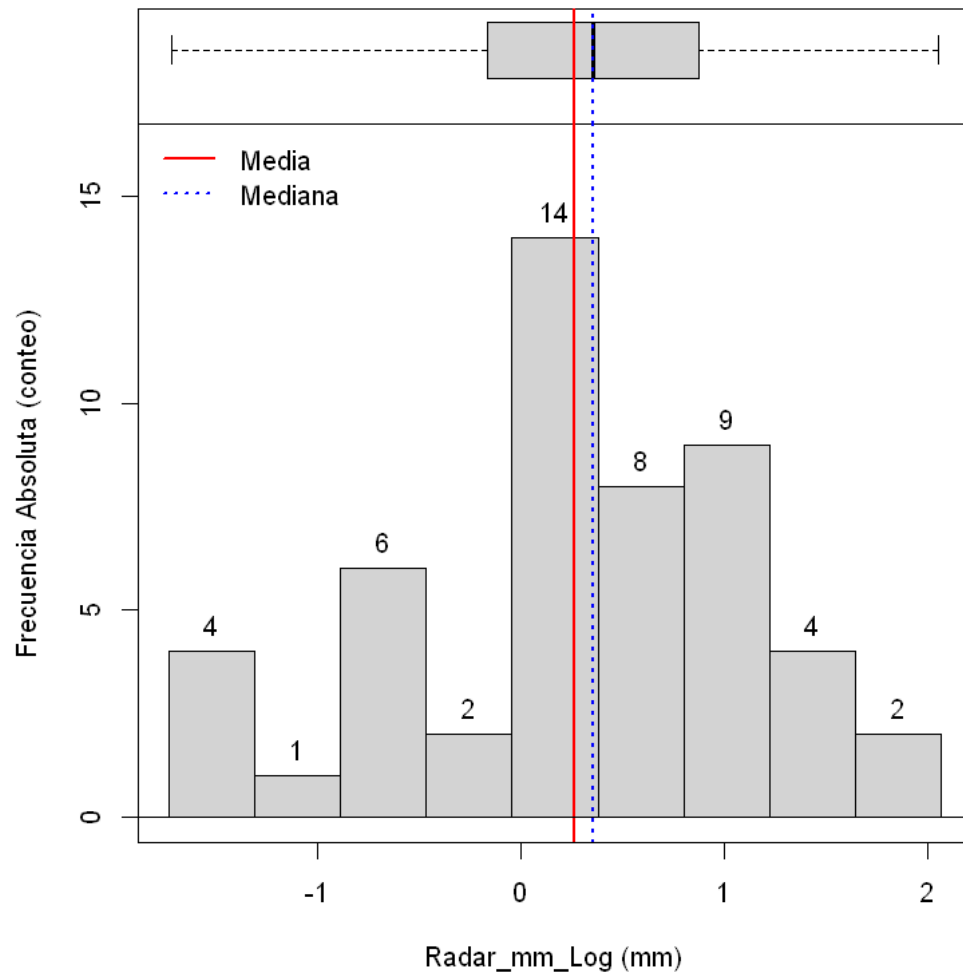
# Estadística basica comparativa entre (Radar_mm), la transformación
→(Radar_mm_Sqrt) y (Radar_mm_Log)
Data_File_Stat <- Val_Estadisticos(Data_File)
Data_File_Stat[,c(3,5,6)]
```

		Radar_mm <dbl>	Radar_mm_Sqrt <dbl>	Radar_mm_Log <dbl>
A data.frame: 13 × 3	No_muestras	50.00000	50.00000	50.00000
	Minimo	0.18000	0.42426	-1.71480
	Cuartil_1er	0.87750	0.93641	-0.13209
	Mediana	1.42500	1.19364	0.35387
	Media	1.83500	1.25106	0.26569
	Cuartil_3er	2.37500	1.54098	0.86467
	Maximo	7.79000	2.79106	2.05284
	Rango	7.61000	2.36679	3.76764
	Rango_Intercuartil	1.49750	0.60456	0.99676
	Varianza	2.47352	0.27536	0.79776
	Desv_Estandar	1.57274	0.52474	0.89317
	Simetria	2.02492	0.74939	-0.46105
	Curtosis	7.99412	3.86947	2.87206

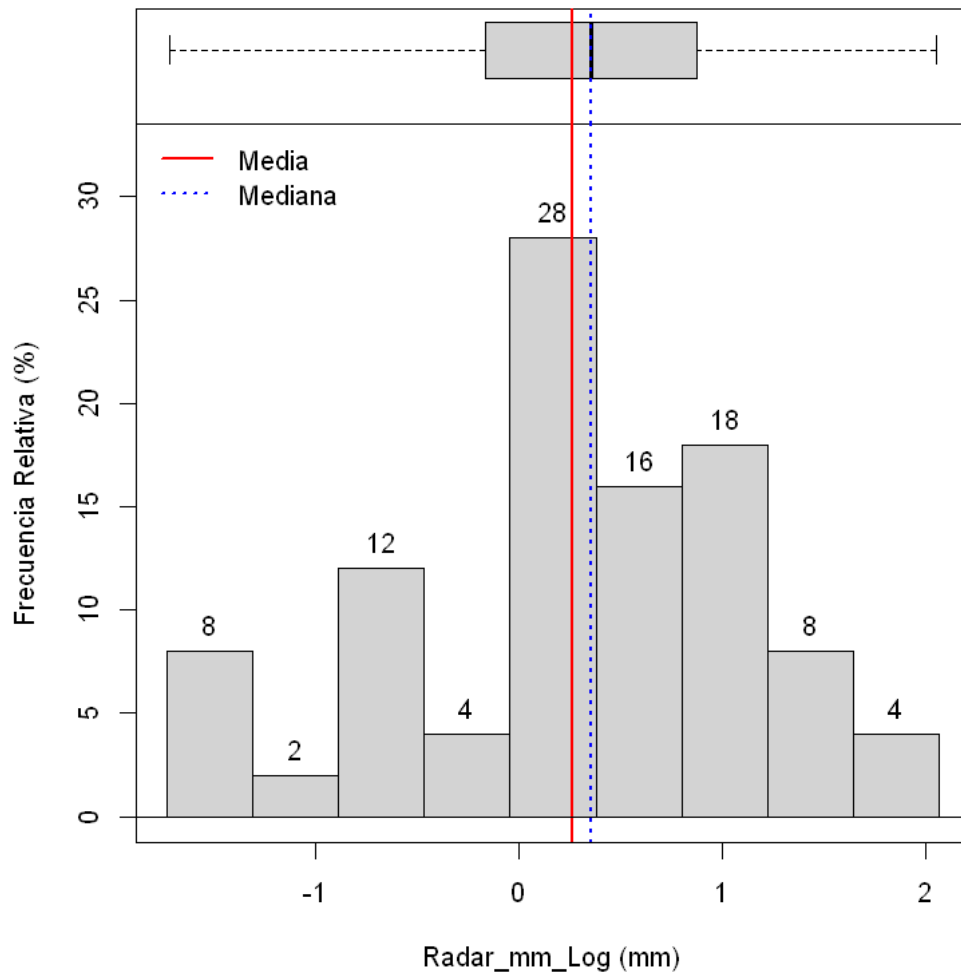
Y graficamos sus respectivos histogramas.

```
[30]: HistBoxplot(x=Radar_mm_Log, mean = Radar_mm_Log_Stat[5,2], median =
→Radar_mm_Log_Stat[4,2], main = "",
xlab = "Radar_mm_Log (mm)", ylab = "Frecuencia Absoluta (conteo)",
→AbsFreq = TRUE, PercentFreq = FALSE,
nbin = 9)
```





```
[31]: HistBoxplot(x=Radar_mm_Log, mean = Radar_mm_Log_Stat[5,2], median = Radar_mm_Log_Stat[4,2], main = "",
  ↳ Radar_mm_Log_Stat[4,2],
  xlab = "Radar_mm_Log (mm)", ylab = "Frecuencia Relativa (%)",
  ↳ AbsFreq = FALSE, PercentFreq = TRUE,
  nbin = 9)
```



La diferencia entre la media y la mediana de la transformación logarítmica es de 0.08818, de forma aparente esta diferencia es mayor comparada con la transformación por raíz cuadrada. El boxplot indica que no existen valores atípicos, sin embargo, se usará la función para detectar los valores atípicos.

```
[32]: Radar_mm_Log_outliers<-OutliersPos(Radar_mm_Log)
      Data_File[Radar_mm_Log_outliers,c(1,2,6)]
```

	UTM_X_m	UTM_Y_m	Radar_mm_Log
	<int>	<int>	<dbl>
A data.frame: 2 × 3	22	462178	2133934
	49	503149	2123937

Como podemos ver, la función ha detectado dos valores atípicos, los cuales retiraremos y obten-

emos sus valores estadísticos.

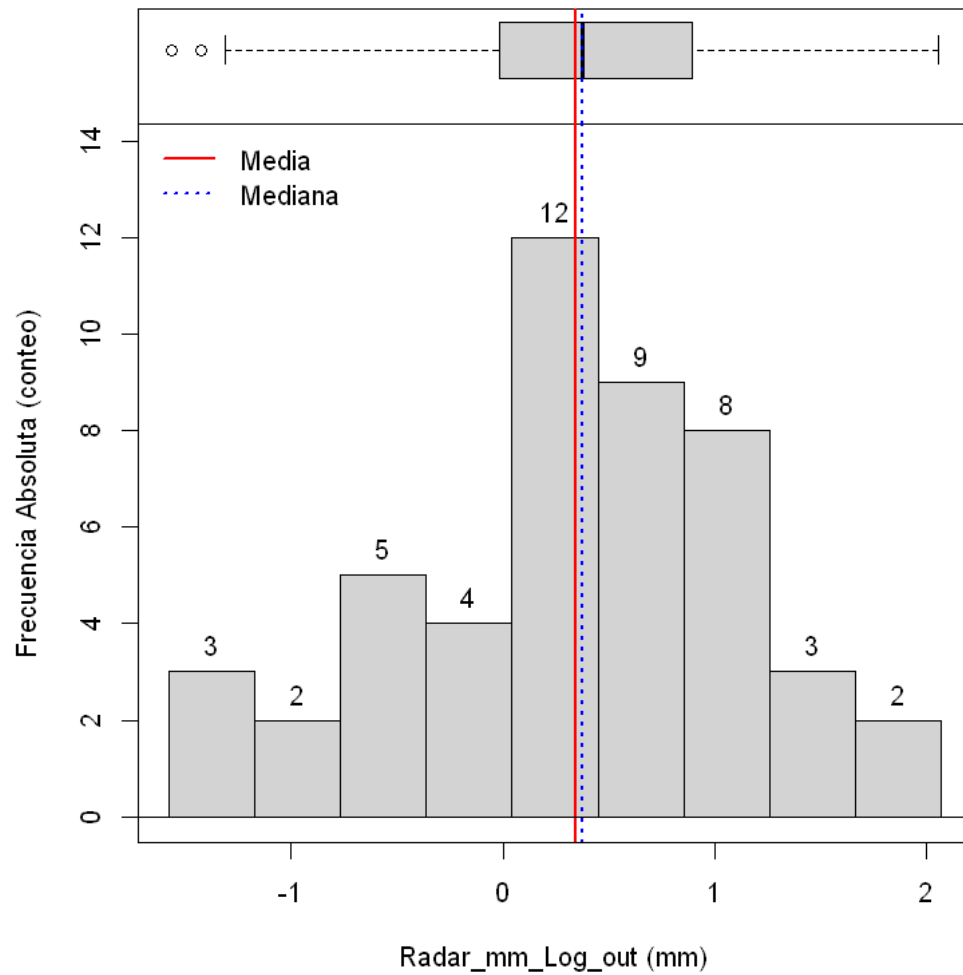
```
[33]: Radar_mm_Log_out<-Radar_mm_Log[-Radar_mm_Log_outliers]

# Estadística básica
Radar_mm_Log_out_Stat<-Estadisticas(Radar_mm_Log_out)
Radar_mm_Log_out_Stat
```

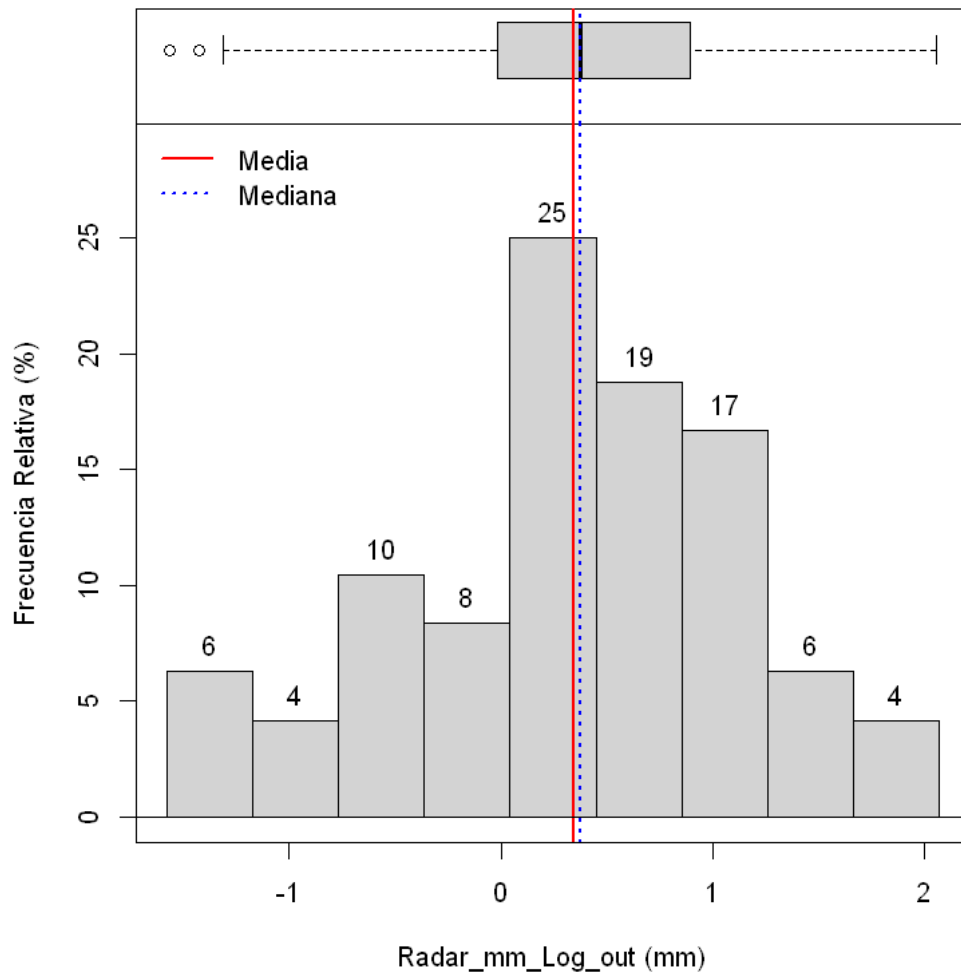
	Statistics <chr>	Values <dbl>
muestras	n	48.0000
minimos	Minimum	-1.5606
cuantiles1	1st. Quartile	-0.0027
medianas	Median	0.3784
medias	Mean	0.3471
cuantiles3	3rd. Quartile	0.8868
maximos	Maximum	2.0528
rangos	Rank	3.6135
rangosInt	Interquartile Rank	0.8895
varianzas	Variance	0.6625
desvs	Standard Deviation	0.8140
CVs	Variation Coeff.	2.3451
simetrias	Skewness	-0.3448
curtosiss	Kurtosis	2.9797

Y graficamos sus histogramas:

```
[34]: HistBoxplot(x=Radar_mm_Log_out, mean = Radar_mm_Log_out_Stat[5,2], median =_
→Radar_mm_Log_out_Stat[4,2], main = "",
      xlab = "Radar_mm_Log_out (mm)", ylab = "Frecuencia Absoluta_
→(conteo)", AbsFreq = TRUE, PercentFreq = FALSE,
      nbin = 9)
```



```
[35]: HistBoxplot(x=Radar_mm_Log_out, mean = Radar_mm_Log_out_Stat[5,2], median = Radar_mm_Log_out_Stat[4,2], main = "",
    xlab = "Radar_mm_Log_out (mm)", ylab = "Frecuencia Relativa (%)",
    AbsFreq = FALSE, PercentFreq = TRUE,
    nbin = 9)
```



La diferencia entre la media y la mediana de la transformación logarítmica quitando los valores atípicos detectados pasó de 0.08818 a 0.0313, sin embargo, el boxplot muestran dos valores atípicos localizados a la izquierda del gráfico. Por lo tanto, retiraremos esos nuevos valores atípicos.

```
[36]: Radar_mm_Log_outliers2<-OutliersPos(Radar_mm_Log_out)
      Radar_mm_Log_out[Radar_mm_Log_outliers2]
```

```
1. -1.42711635564015 2. -1.56064774826467
```

Y hacemos un analisis estadístico de este nuevo conjunto de valores.

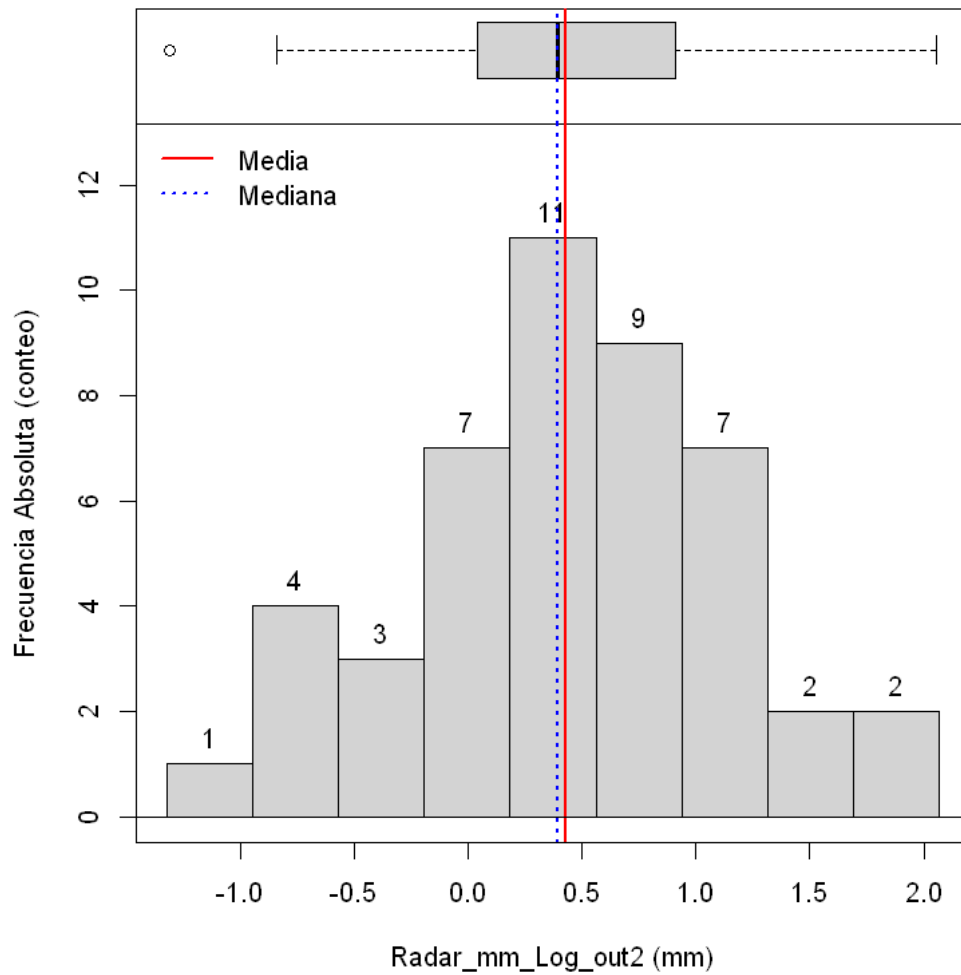
```
[37]: # Variable without distributional outliers 2
      Radar_mm_Log_out2<-Radar_mm_Log_out[-Radar_mm_Log_outliers2]
```

```
#2º analisis sin valores atipicos
# Estadística básica
Radar_mm_Log_out2_Stat<-Estadisticas(Radar_mm_Log_out2)
Radar_mm_Log_out2_Stat
```

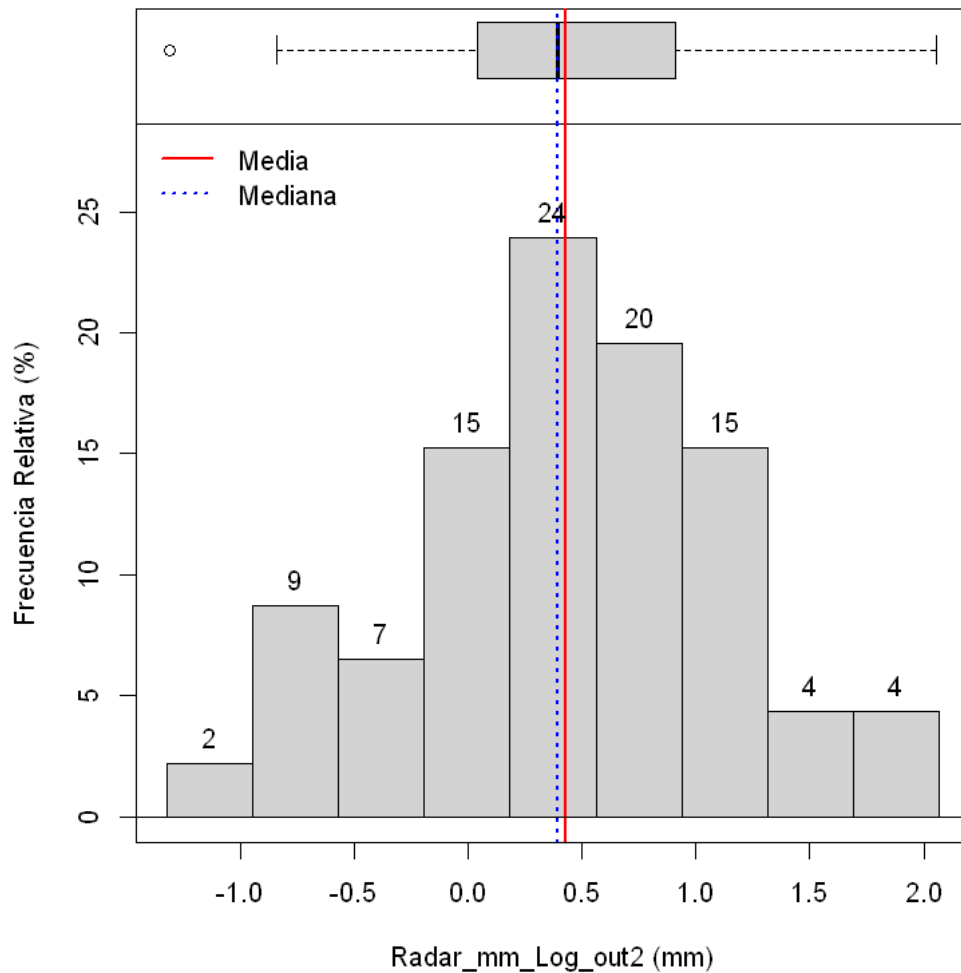
	Statistics <chr>	Values <dbl>
muestras	n	46.0000
minimos	Minimum	-1.3093
cuantiles1	1st. Quartile	0.0440
medianas	Median	0.3920
medias	Mean	0.4271
cuantiles3	3rd. Quartile	0.9011
maximos	Maximum	2.0528
rangos	Rank	3.3622
rangosInt	Interquartile Rank	0.8571
varianzas	Variance	0.5346
desvs	Standard Deviation	0.7312
CVs	Variation Coeff.	1.7118
simetrias	Skewness	-0.1103
curtosiss	Kurtosis	2.9190

Y graficamos su respectivo histograma.

```
[38]: HistBoxplot(x=Radar_mm_Log_out2, mean = Radar_mm_Log_out2_Stat[5,2], median =_
→Radar_mm_Log_out2_Stat[4,2], main = "",
      xlab = "Radar_mm_Log_out2 (mm)", ylab = "Frecuencia Absoluta",
→(conteo)", AbsFreq = TRUE, PercentFreq = FALSE,
      nbin = 9)
```



```
[39]: HistBoxplot(x=Radar_mm_Log_out2, mean = Radar_mm_Log_out2_Stat[5,2], median =
→Radar_mm_Log_out2_Stat[4,2], main = "",
        xlab = "Radar_mm_Log_out2 (mm)", ylab = "Frecuencia Relativa (%)",
→AbsFreq = FALSE, PercentFreq = TRUE,
        nbin = 9)
```



La diferencia entre la media y la mediana de la transformación logarítmica quitando los valores atípicos detectados pasó de 0.0313 a 0.0351, es decir, aumentó el valor, sin ser significativo este aumento, con lo cual podemos considerar que hacer el segundo retiro de valores atípicos es innecesario.

### 3.1.2 Análisis estadístico univariado para los datos obtenidos de pluviómetros (Pluv\_mm).

Ahora se hace el mismo análisis estadístico a los datos obtenidos de los pluviómetros (Pluv\_mm). Empezamos obteniendo los valores estadísticos.

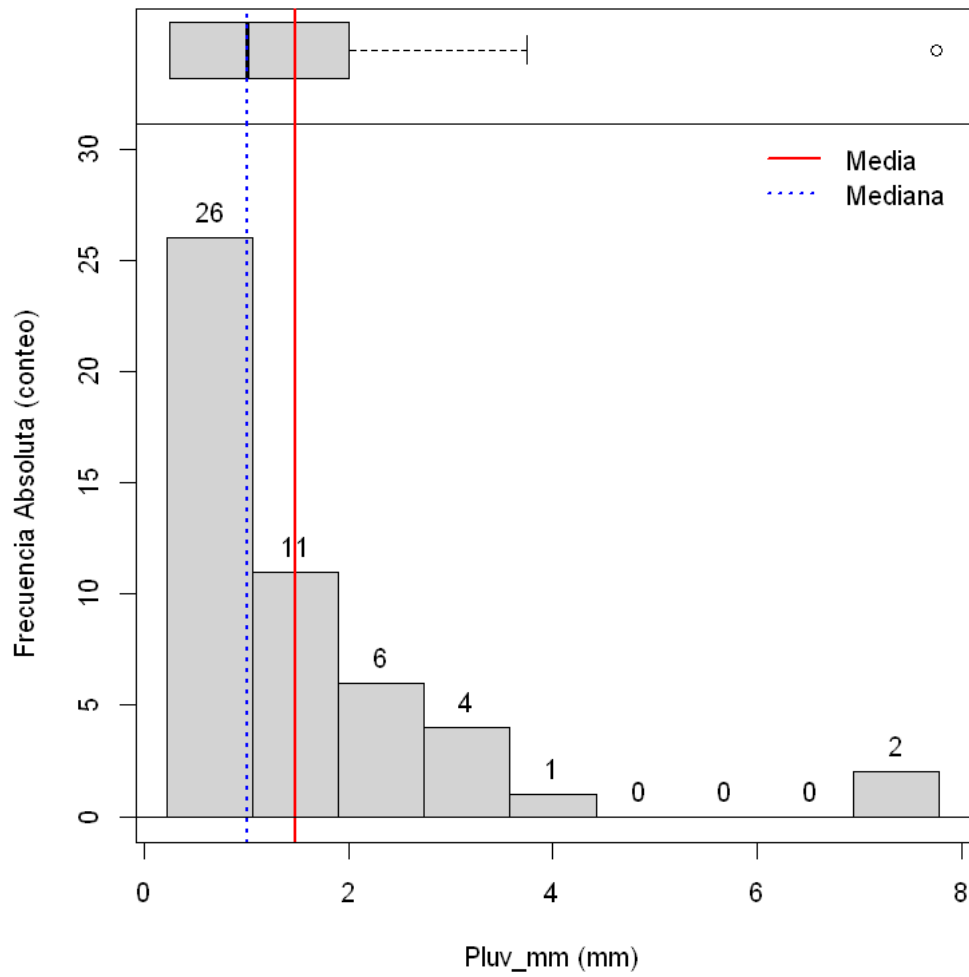
```
[40]: Pluv_mm_Stat <- Estadisticas(Pluv_mm)
      Pluv_mm_Stat
```



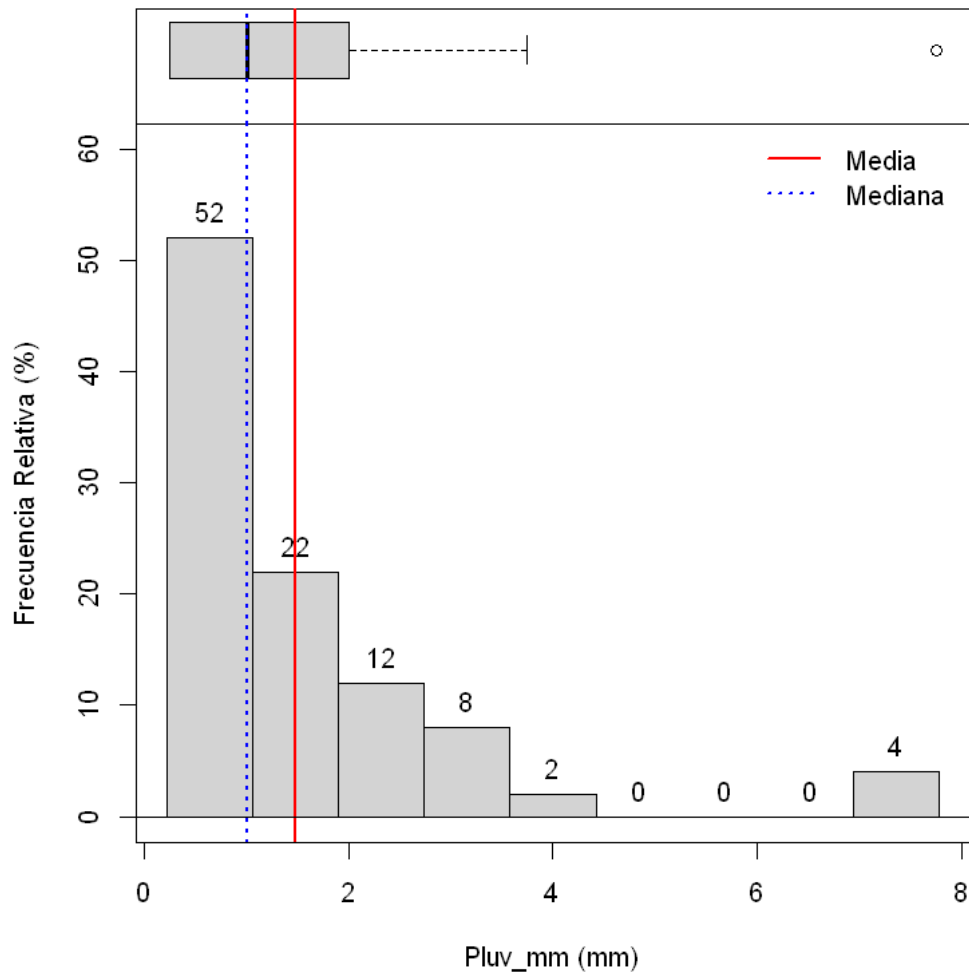
	Statistics <chr>	Values <dbl>
muestras	n	50.0000
minimos	Minimum	0.2500
cuantiles1	1st. Quartile	0.3125
medianas	Median	1.0000
medias	Mean	1.4700
cuantiles3	3rd. Quartile	1.9375
maximos	Maximum	7.7500
rangos	Rank	7.5000
rangosInt	Interquartile Rank	1.6250
varianzas	Variance	2.5756
desvs	Standard Deviation	1.6049
CVs	Variation Coeff.	1.0917
simetrias	Skewness	2.4693
curtosiss	Kurtosis	10.0542

Y su respectivo histograma.

```
[41]: HistBoxplot(x=Pluv_mm, mean = Pluv_mm_Stat[5,2], median = Pluv_mm_Stat[4,2],
  ↳main="",
      xlab = "Pluv_mm (mm)", ylab = "Frecuencia Absoluta (conteo)",
  ↳AbsFreq = TRUE, PercentFreq = FALSE,
      nbin = 9)
```



```
[42]: HistBoxplot(x=Pluv_mm, mean = Pluv_mm_Stat[5,2], median = Pluv_mm_Stat[4,2],
  ↳main="",
      xlab = "Pluv_mm (mm)", ylab = "Frecuencia Relativa (%)", AbsFreq =
  ↳FALSE, PercentFreq = TRUE,
      nbin = 9)
```



Podemos notar que la diferencia entre la media y la mediana es de 0.47, lo cual nos indica que la variable es asimétrica positiva y el histograma nos confirma esta información.

### Transformación logarítmica.

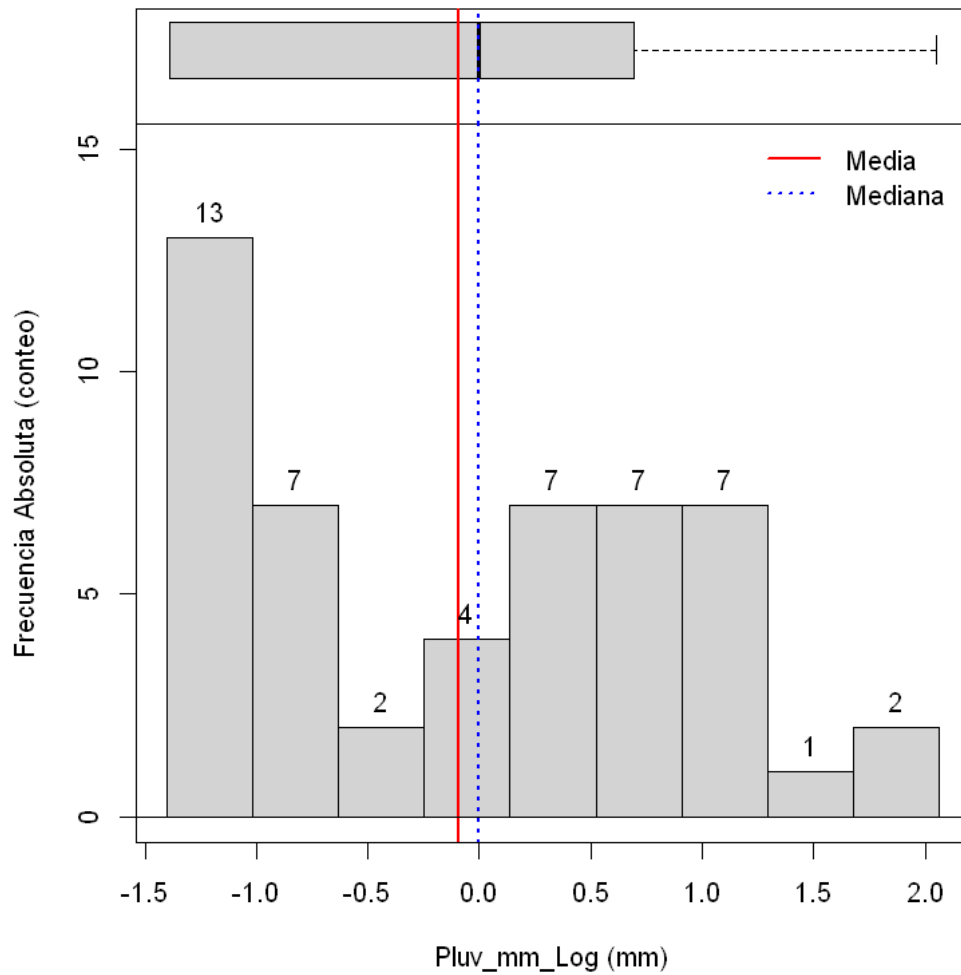
Analizando el histograma podemos notar un fuerte crecimiento en la parte izquierda, por lo tanto usaremos la transformación logarítmica.

```
[43]: Pluv_mm_Log <- log(Pluv_mm)
      Pluv_mm_Log_Stat <- Estadisticas(Pluv_mm_Log)
      Pluv_mm_Log_Stat
```

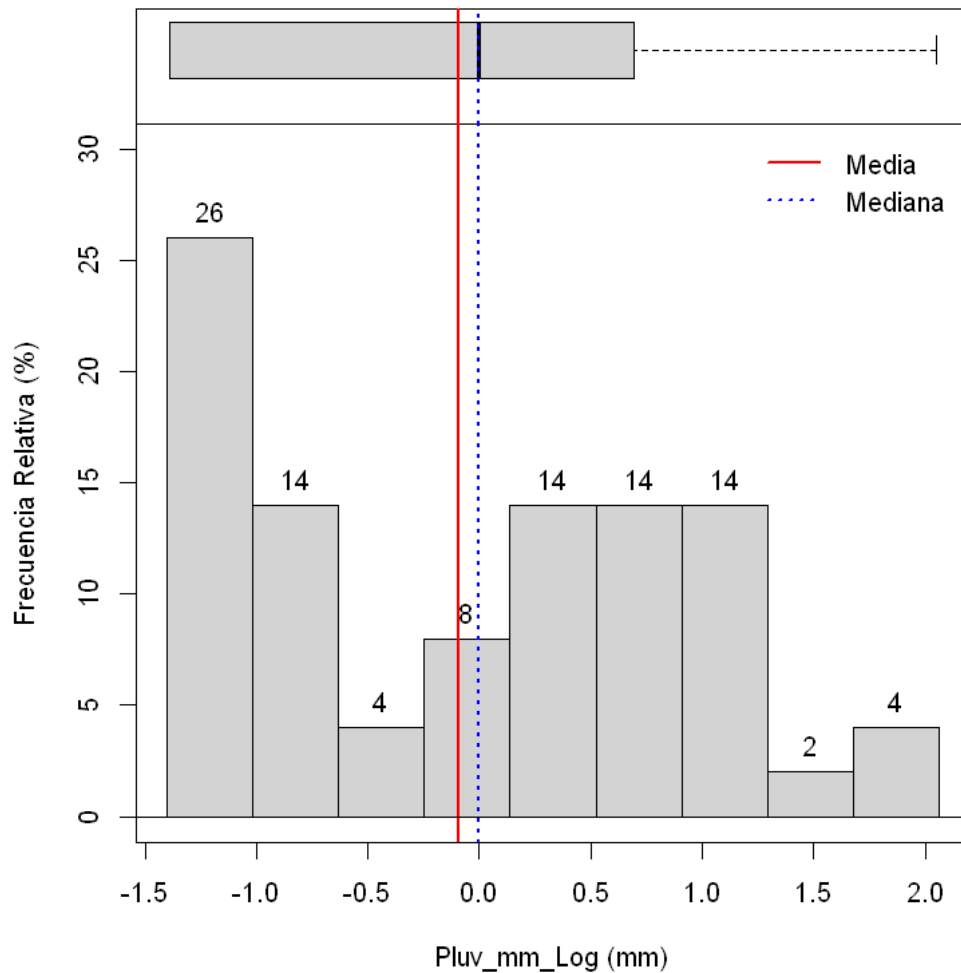
	Statistics <chr>	Values <dbl>
muestras	n	50.0000
minimos	Minimum	-1.3863
cuantiles1	1st. Quartile	-1.2130
medianas	Median	0.0000
medias	Mean	-0.0903
cuantiles3	3rd. Quartile	0.6598
maximos	Maximum	2.0477
rangos	Rank	3.4340
rangosInt	Interquartile Rank	1.8728
varianzas	Variance	1.0004
desvs	Standard Deviation	1.0002
CVs	Variation Coeff.	-11.0754
simetrias	Skewness	0.1124
curtosiss	Kurtosis	2.0120

Los histogramas de la transformación logarítmica son los siguientes:

```
[44]: HistBoxplot(x=Pluv_mm_Log, mean = Pluv_mm_Log_Stat[5,2], median =
→Pluv_mm_Log_Stat[4,2], main = "",
      xlab = "Pluv_mm_Log (mm)", ylab = "Frecuencia Absoluta (conteo)",
→AbsFreq = TRUE, PercentFreq = FALSE,
      nbin = 9)
```



```
[45]: HistBoxplot(x=Pluv_mm_Log, mean = Pluv_mm_Log_Stat[5,2], median = Pluv_mm_Log_Stat[4,2], main = "",
    xlab = "Pluv_mm_Log (mm)", ylab = "Frecuencia Relativa (%)", AbsFreq = FALSE, PercentFreq = TRUE,
    nbin = 9)
```



La diferencia entre la media y la mediana pasó de 0.47 a 0.09, lo cual es bajo. El boxplot no muestra valores atípicos, sin embargo, nos aseguraremos que esto sea cierto usando la función “OutliersPos”.

```
[46]: Pluv_mm_Log_outliers<-OutliersPos(Pluv_mm_Log)
      print(Pluv_mm_Log_outliers)
```

```
numeric(0)
```

No se encontraron valores atípicos, por lo tanto, aquí se termina el análisis univariado de esta variable.

### 3.2 Análisis estadístico bivariado.

Como pudimos notar durante el análisis exploratorio univariado, necesitamos de dos elementos para interpretar las características estadísticas de una variable: un histograma y una tabla con los valores estadísticos. Con el caso del análisis exploratorio bivariado necesitamos un diagrama de dispersión o scatterplot y los grados de dependencia.

Un diagrama de dispersión es una gráfica compuesta por pares de valores de dos variables aleatorias  $(x_i, y_i)$ .

Los grados de dependencia se miden usando el coeficiente de correlación lineal de Pearson:

$$\rho_{XY} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}}$$

El coeficiente de correlación de Spearman:

$$\rho = 1 - \frac{6 \sum D^2}{N(N^2 - 1)}$$

Y el coeficiente de correlación de Kendall:

$$\tau = \frac{u - \bar{u}}{\binom{n}{2}}$$

#### 3.2.1 Cálculo de grados de dependencia

Para calcular estos grados de dependencia lo hacemos de la siguiente forma:

```
[47]: cor(Radar_mm , Pluv_mm, method = "pearson")
```

0.941048488234573

```
[48]: cor(Radar_mm , Pluv_mm, method = "spearman")
```

0.848045300430134

```
[49]: cor(Radar_mm , Pluv_mm, method = "kendall")
```

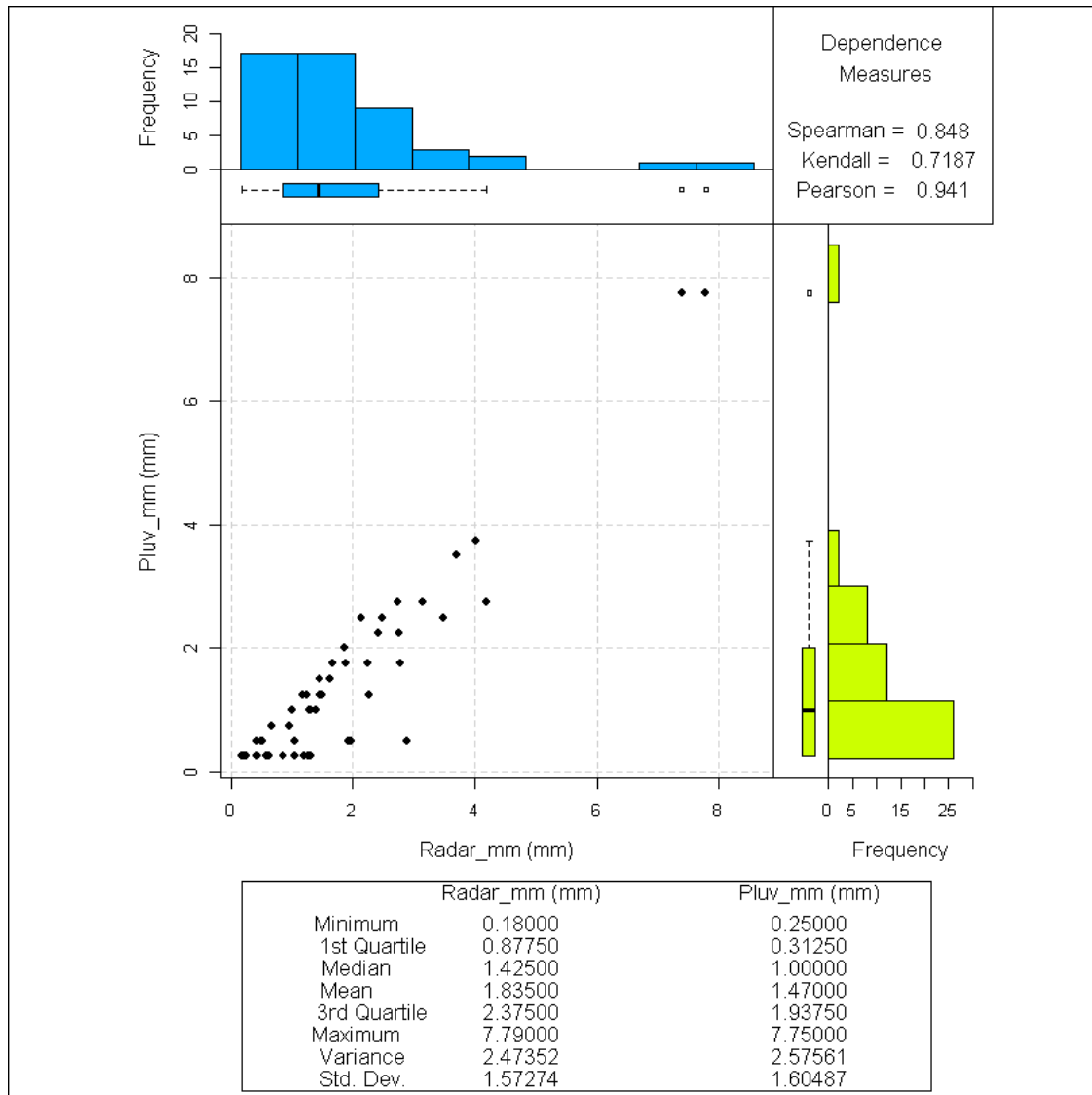
0.718715084238533

Respecto al coeficiente de Pearson, su valor es de 0.9410, lo cual podemos considerar como cuasi-lineal. Sin embargo, los valores de la correlación de Spearman (0.8480) y Kendall (0.7187) indican que el modelo no es cuasi-lineal, por lo que es posible que la dependencia lineal esté alterada por valores atípicos.

#### 3.2.2 Diagrama de dispersión.

El diagrama de dispersión se grafica de la siguiente manera:

```
[50]: ScatterPlot(Radar_mm , Pluv_mm, 9,
                  Xmin = Radar_mm_Stat[2,2], Xmax = Radar_mm_Stat[7,2],
                  Ymin = Pluv_mm_Stat[2,2], Ymax = Pluv_mm_Stat[7,2],
                  XLAB = "Radar_mm (mm)", YLAB = "Pluv_mm (mm)")
```



Podemos notar en el gráfico de dispersión que hay dos pares atípicos localizados en la esquina superior derecha, los cuales podrían ser la causa de la cuasi-linearidad que indica la medida de dependencia de Pearson. Estos valores coinciden con los valores atípicos que se detectaron durante el análisis univariado. Sin embargo, es importante señalar que los valores atípicos encontrados en el análisis exploratorio univariado no son necesariamente valores atípicos en el análisis exploratorio bivariado, se debe evaluar la conveniencia de retirar cada par.

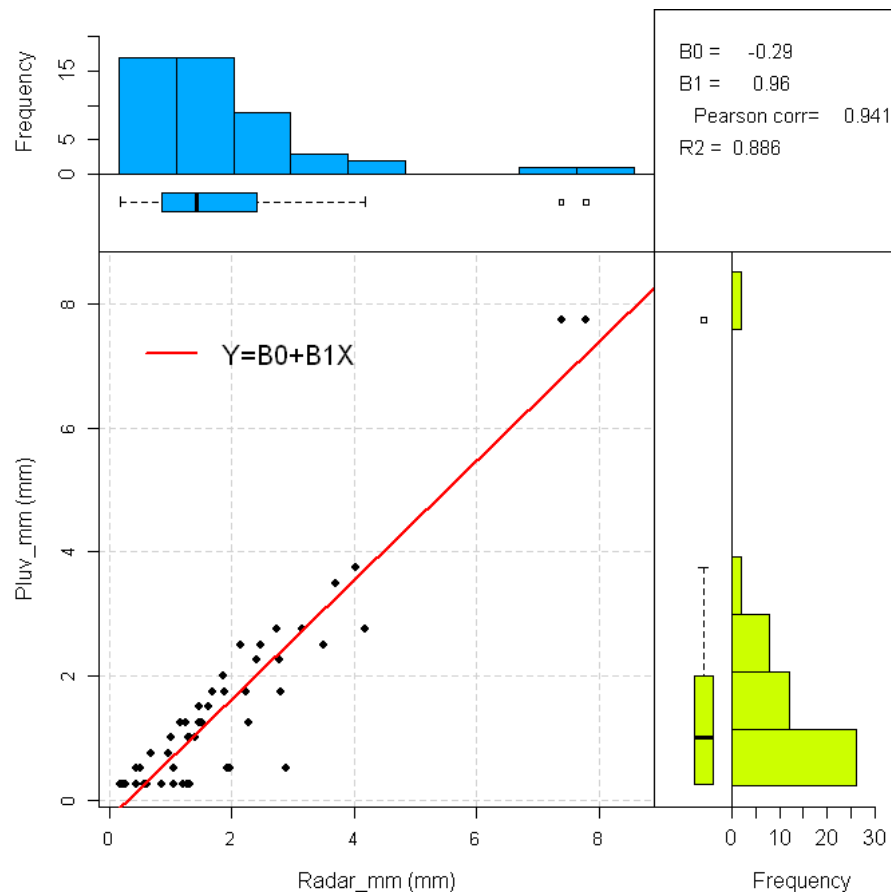


### 3.2.3 Analisis de regresión lineal.

Como se mencionó en clase, la regresión trata de establecer relaciones funcionales entre variables aleatorias, en este caso, la relación se establece con una recta. Para hacer el análisis necesitamos de los parámetros de la recta y el análisis de residuos.

El grafico de dispersión con línea de regresión se genera de la siguiente forma:

```
[51]: scatterplotReg(Radar_mm , Pluv_mm, 9,  
    Xmin = Radar_mm_Stat[2,2], Xmax = Radar_mm_Stat[7,2],  
    Ymin = Pluv_mm_Stat[2,2], Ymax = Pluv_mm_Stat[7,2],  
    XLAB = "Radar_mm (mm)", YLAB = "Pluv_mm (mm)")
```



De este grafico nos interesa saber los valores de la regresión lineal y su error cuadrático. Para hacer la regresión lineal usamos la función "linear\_regression":

```
[52]: # Linear Regression
X<-Radar_mm
Y<-Pluv_mm

linear_regression <-lm(Y ~ X)

# Linear Regression Parameters
B0 <- linear_regression$coefficients[1]
B0
B1 <- linear_regression$coefficients[2]
B1
```

**(Intercept):** -0.292098951631834

**X:** 0.960271908246231

Ya que tenemos los parámetros de la recta  $B_0$  y  $B_1$ , hacemos el calculo de los residuos.

```
[53]: # Regression line and Residual Calculation
Y_Regression <- linear_regression$fitted.values
Y_Residual <- linear_regression$residuals
```

Ya que tenemos calculados los residuos necesitamos obtener sus valores estadísticos.

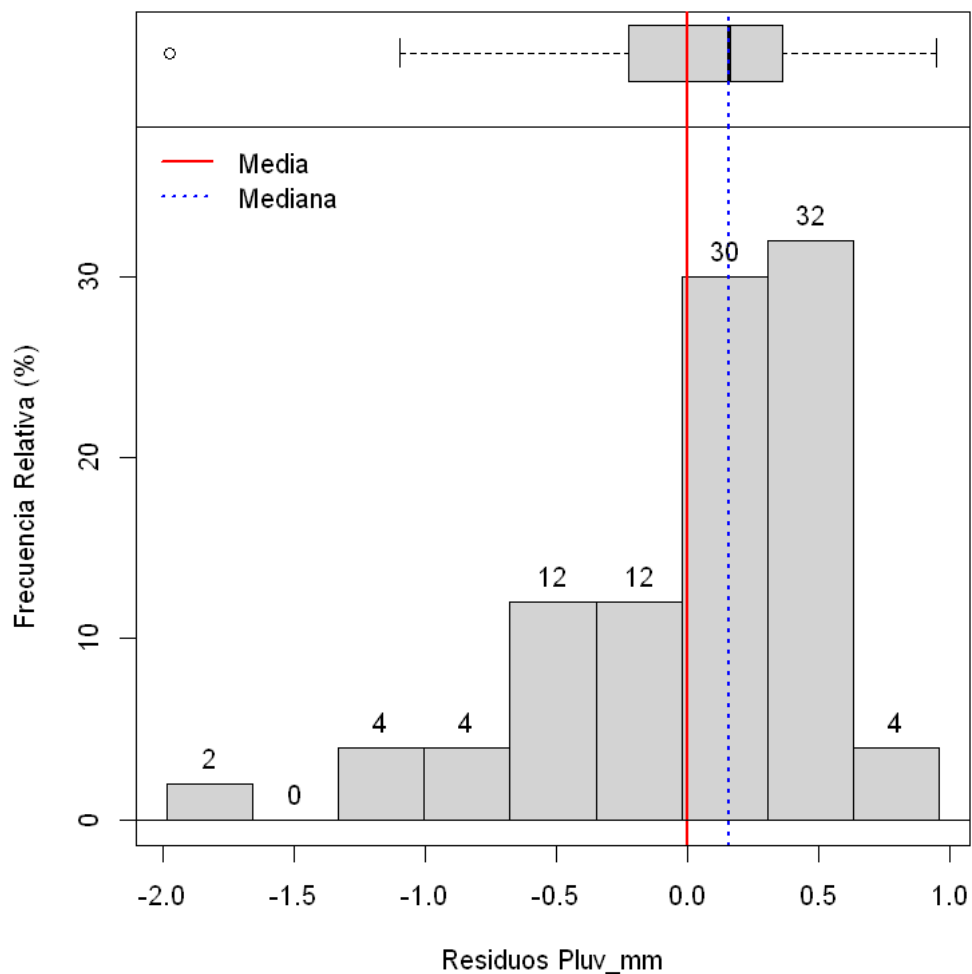
```
[54]: Y_Residual_Stat<-Estadisticas(Y_Residual)
write.csv(Y_Residual_Stat , file = "Results/AED/Pluv_mm_Residual_Stat.csv")
Y_Residual_Stat
```

	Statistics <chr>	Values <dbl>
muestras	n	5.000000e+01
minimos	Minimum	-1.973500e+00
cuantiles1	1st. Quartile	-1.988000e-01
medianas	Median	1.562000e-01
medias	Mean	0.000000e+00
cuantiles3	3rd. Quartile	3.552000e-01
maximos	Maximum	9.457000e-01
rangos	Rank	2.919200e+00
rangosInt	Interquartile Rank	5.540000e-01
varianzas	Variance	2.947000e-01
desvs	Standard Deviation	5.429000e-01
CVs	Variation Coeff.	-2.318152e+17
simetrias	Skewness	-1.347400e+00
curtosiss	Kurtosis	5.138200e+00

Y tambien necesitamos obtener el histograma de estos residuos.

```
[55]: HistBoxplot(x=Y_Residual, mean = Y_Residual_Stat[5,2], median =
→Y_Residual_Stat[4,2], main = "",
```

```
xlab = "Residuos Pluv_mm", ylab = "Frecuencia Relativa (%)", AbsFreq =
→ FALSE, PercentFreq = TRUE )
```



Si analizamos los valores estadísticos y el histograma de los residuos, podemos notar que el valor esperado es de 0, su varianza es de 0.2947 y la diferencia entre la media y la mediana es de 0.1562, lo cual nos indica que tiene asimetría negativa, por lo tanto, los residuos no cumplen con todas las condiciones que demanda la regresión lineal.

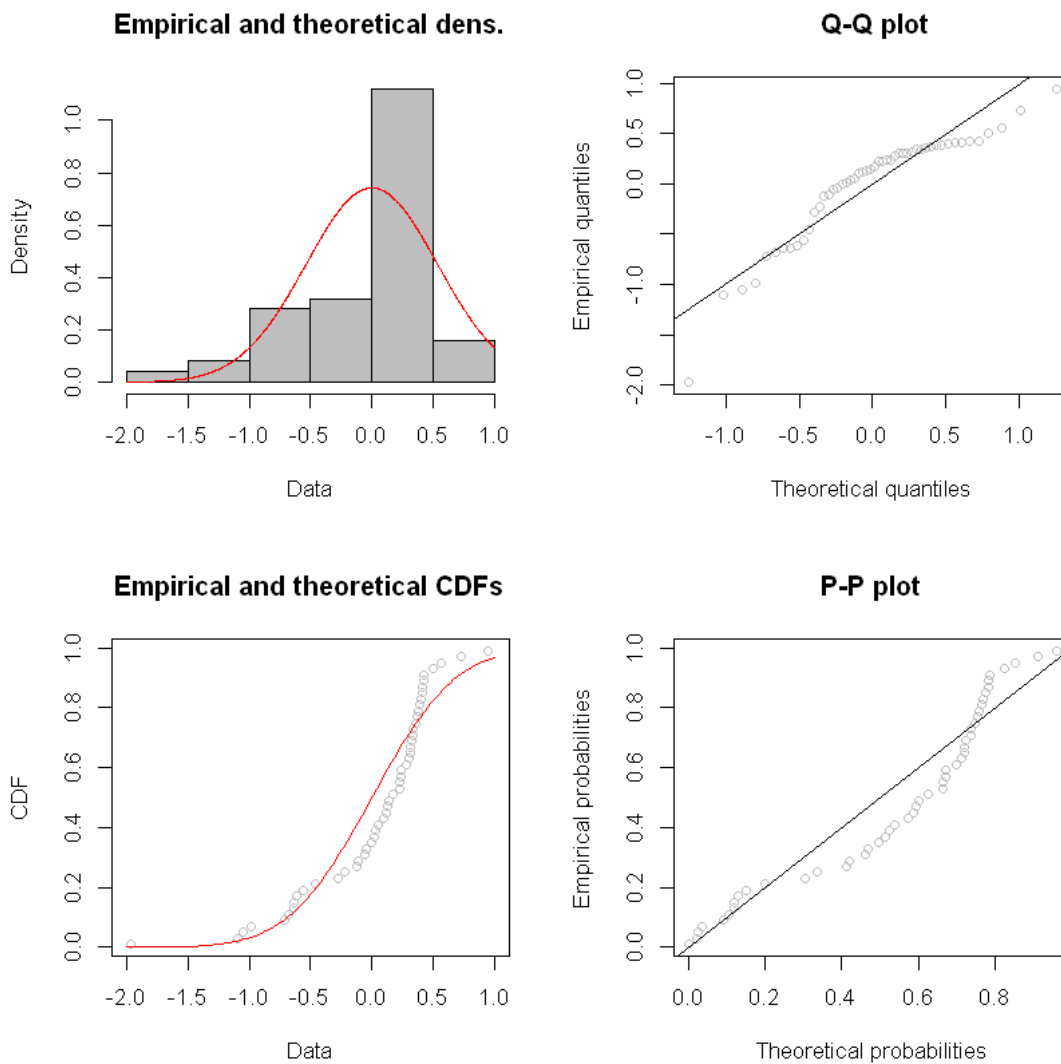
```
[56]: FitDistr2_Residual_normal<-FitDistribution(data = Y_Residual, DISTR="norm",
→BREAKS = "Sturges", col = "gray", DistName = "Normal")
```

```
$start.arg
$start.arg$mean
[1] -2.341877e-18
```

```
$start.arg$sd  
[1] 0.5374263
```

```
$fix.arg  
NULL
```

```
Warning message in hist.default(data, breaks = breaks, plot = FALSE, ...):  
"argument 'col' is not made use of"
```



Para confirmar que los residuos no cumplen con las condiciones de la regresión lineal podemos sobreponer el histograma con la distribución normal (figura superior izquierda), ahí podemos ver

que una de las barras del histograma sobrepasa a la función de distribución. El grafico Q-Q plot (figura superior derecha) también muestra que solo unas pocas muestras están en la recta. La grafica comparativa entre las funciones de distribución acumulativas empírica y teórica (figura inferior izquierda) no muestran un buen ajuste Y en el caso del grafico P-P plot (figura inferior derecha) solo las muestras de la esquina inferior izquierda y unas pocas de la parte central se posicionan cerca de la recta.

Ahora debemos aplicar un test de normalidad, en este caso tenemos dos opciones: hipótesis de Kolmogorov-Smirnov y la hipótesis de Anderson-Darling.

La hipótesis de Kolmogorov-Smirnov se usa para contrastar la hipótesis de normalidad, el estadístico de prueba es la máxima diferencia:

$$D = \max |F_n(x) - F_{va}(x)|$$

Donde  $F_n(x)$  es la función de distribución paramétrica, en este caso la función normal. Y  $F_{va}(x)$  es la función de la variable aleatoria.

La hipótesis de Anderson-Darling es una prueba no paramétrica que se basa en la comparación de las muestras Y y la función de distribución de probabilidad teórica F. Su fórmula es:

$$S = \sum_{k=1}^N \frac{2k-1}{N} [\ln(F(Y_k)) + \ln(1 - F(Y_{N+1-k}))]$$

El valor p es una probabilidad que mide la evidencia en contra de la hipótesis nula. Un valor p más pequeño proporciona una evidencia más fuerte en contra de la hipótesis nula. esta hipótesis se usa para determinar si los datos siguen una distribución normal.

Si  $p \leq \alpha$  donde  $\alpha$  es el nivel de significancia la decisión es rechazar la hipótesis nula y concluir que sus datos no siguen una distribución normal.

Si  $p > \alpha$  donde  $\alpha$  es el nivel de significancia la decisión es no rechazar la hipótesis nula y concluir que sus datos no tiene suficiente evidencia para concluir que los datos no siguen una distribución normal.

```
[57]: FD_HT_Residual_normal<-FitDistr2_Residual_normal$x
      FD_HT_Residual_normal
```

	Nombre	Nivel de significancia	P-valor	Estadístico	Decisión
A data.frame: 2 × 5	<chr>	<chr>	<chr>	<chr>	<chr>
	Kolmogorov-Smirnov	0.05	0.136	0.1605	No rechazo H0
	Anderson-Darling	0.05	0.07826	2.13	No rechazo H0

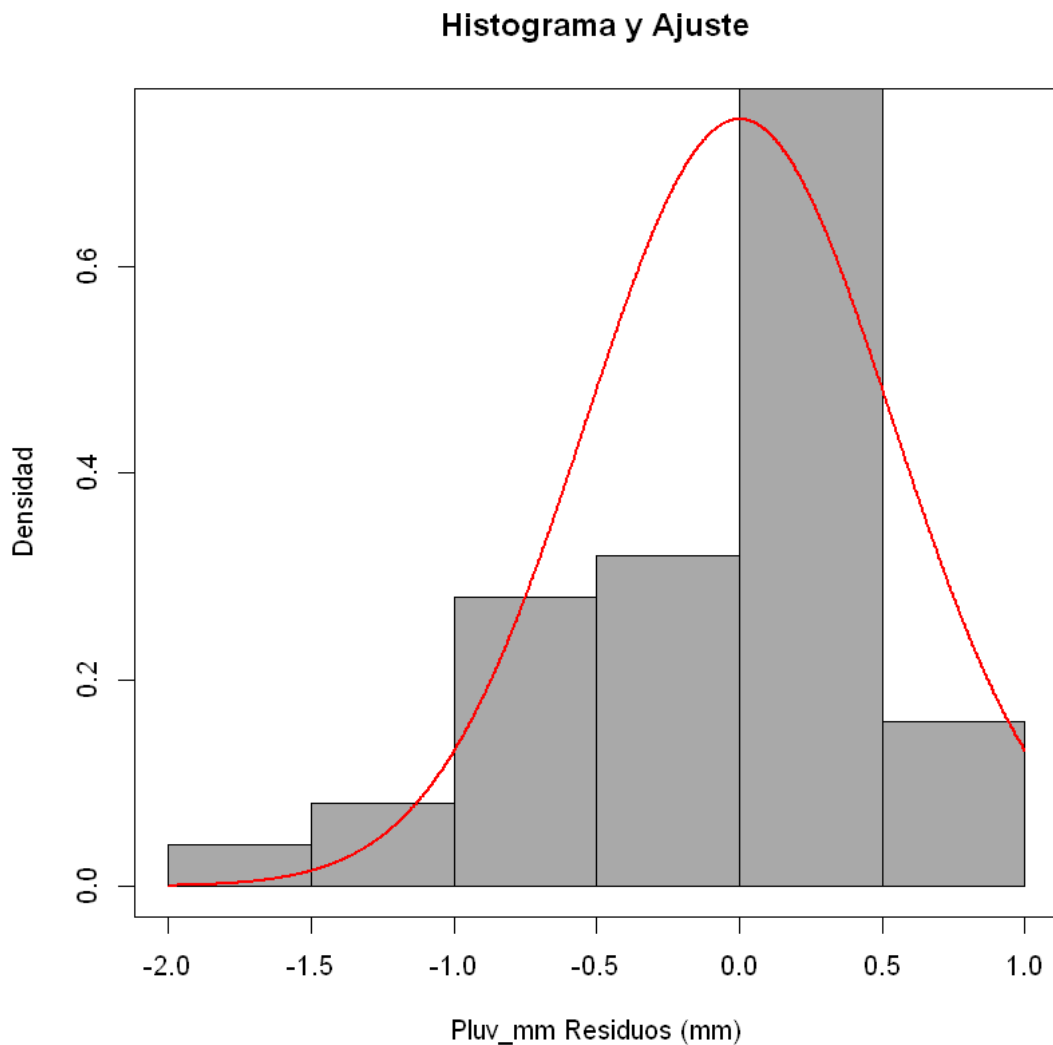
Bajo la prueba de Kolmogorov-Smirnov podemos ver que la normalidad de los residuos es de no rechazo. Mientras que la prueba de Anderson-Darling no rechaza la hipótesis de normalidad. Sin embargo, los valores obtenidos de estas pruebas son muy cercanos al nivel de significancia, por lo que podríamos reforzar la evidencia de no normalidad de la variable.

```
[58]: FD_FP_Residual_normal<-FitDistr2_Residual_normal$y
      FD_FP_Residual_normal
```

		Normal <dbl>
A data.frame: 4 × 1	Media	-2.341877e-18
	Desviación estándar	5.374263e-01
	Máxima Verosimilitud	-3.989875e+01
	AICC	8.379750e+01

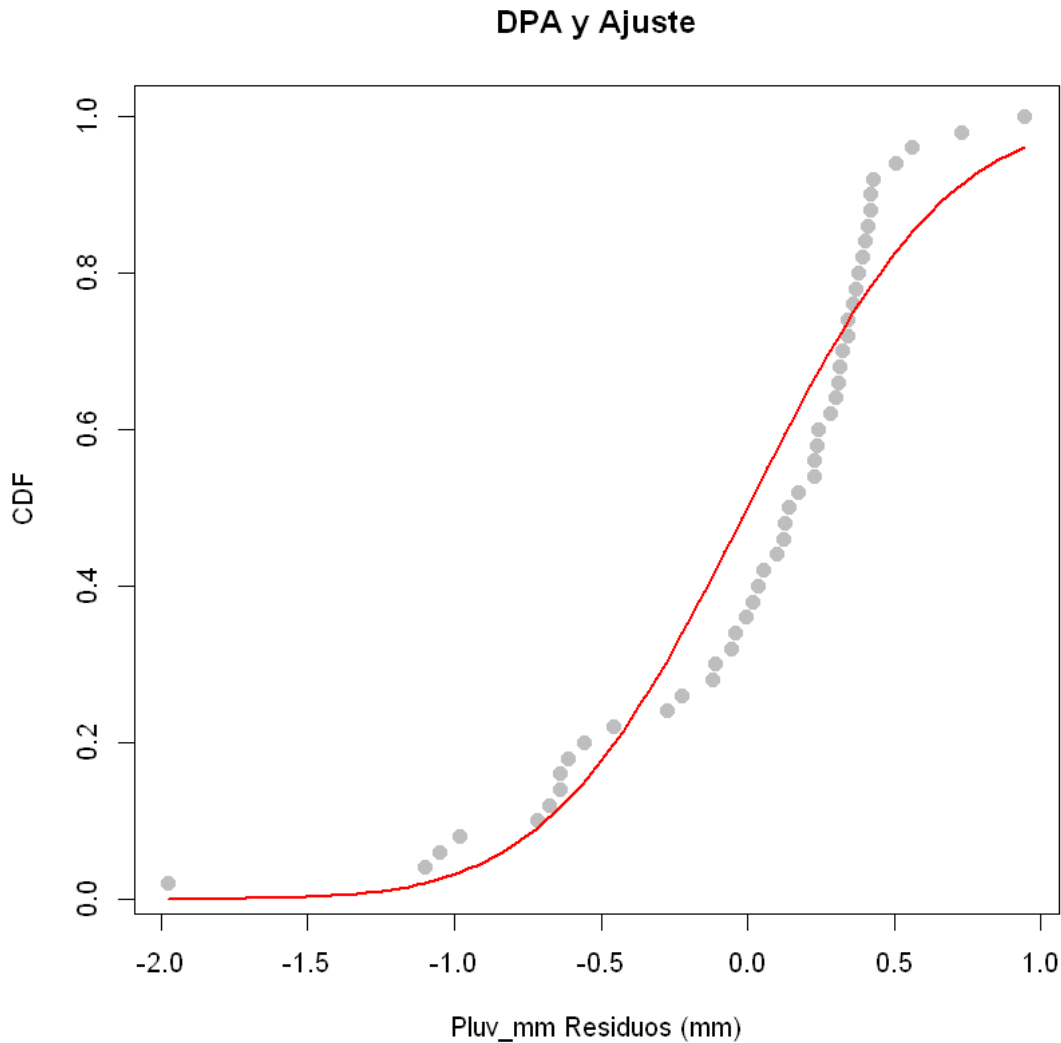
Los siguientes gráficos son los mismos que se analizaron en el vector "Fit-Distr2\_Residual\_normal", tenemos el histograma la función de distribución normal.

```
[59]: PARA_Residual_normal <- list(mean = as.numeric(FD_FP_Residual_normal[1,1]), sd =
  ↪ as.numeric(FD_FP_Residual_normal[2,1]))
HistModel(x = Y_Residual, distr = "norm", para = PARA_Residual_normal, breaks =
  ↪ "Sturges", freq = FALSE, main = "Histograma y Ajuste", xlab = "Pluv_mm Residuos",
  ↪ "(mm)",
  ylab = "Densidad", colCurve = "red", col = "darkgray")
```



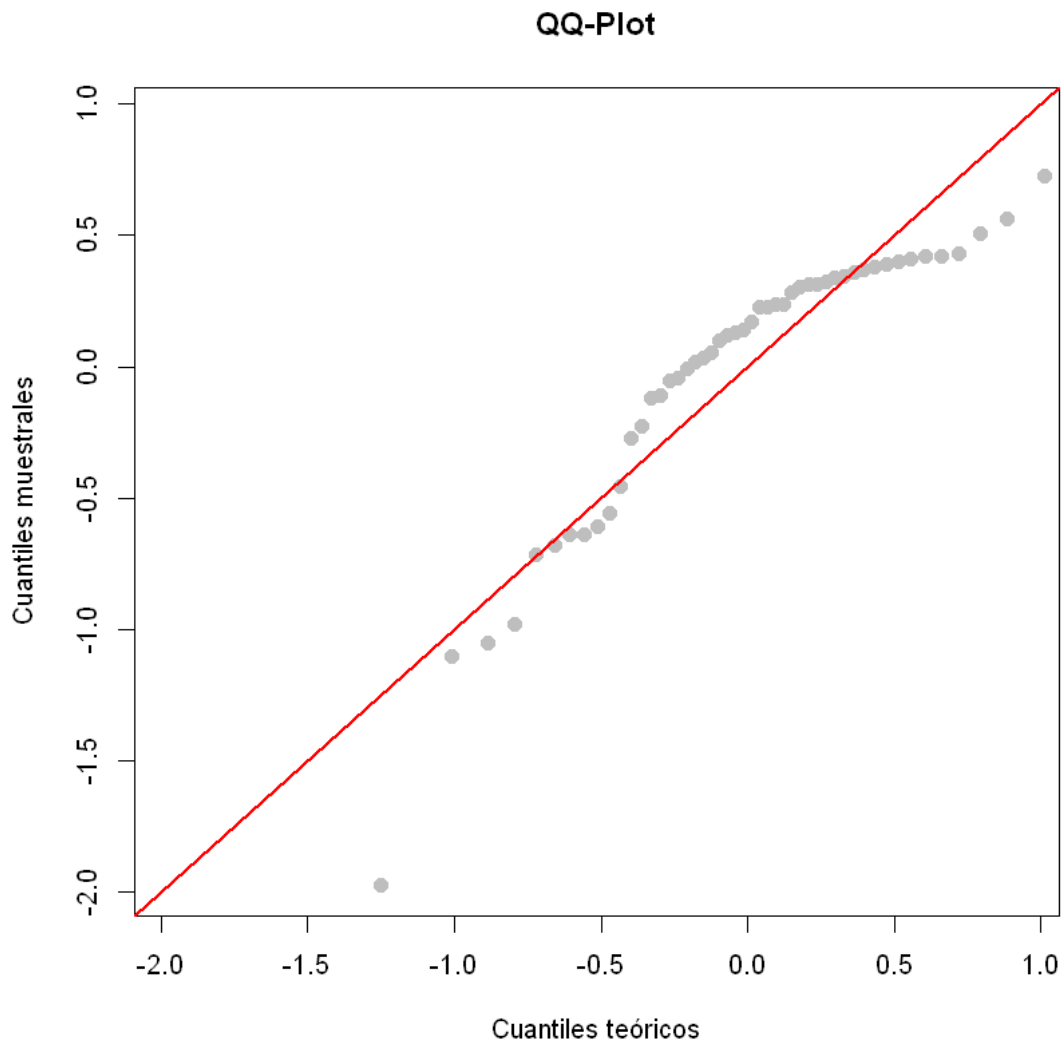
Graficamos su función de distribución acumulativa.

```
[60]: CDF(x = Y_Residual, distr = "norm", para = PARA_Residual_normal, col = "gray",  
→main = "DPA y Ajuste", xlab = "Pluv_mm Residuos (mm)",  
lcol = "red", lwd = 2)
```



El Grafico cuantil-cuantil (Q-Q plot).

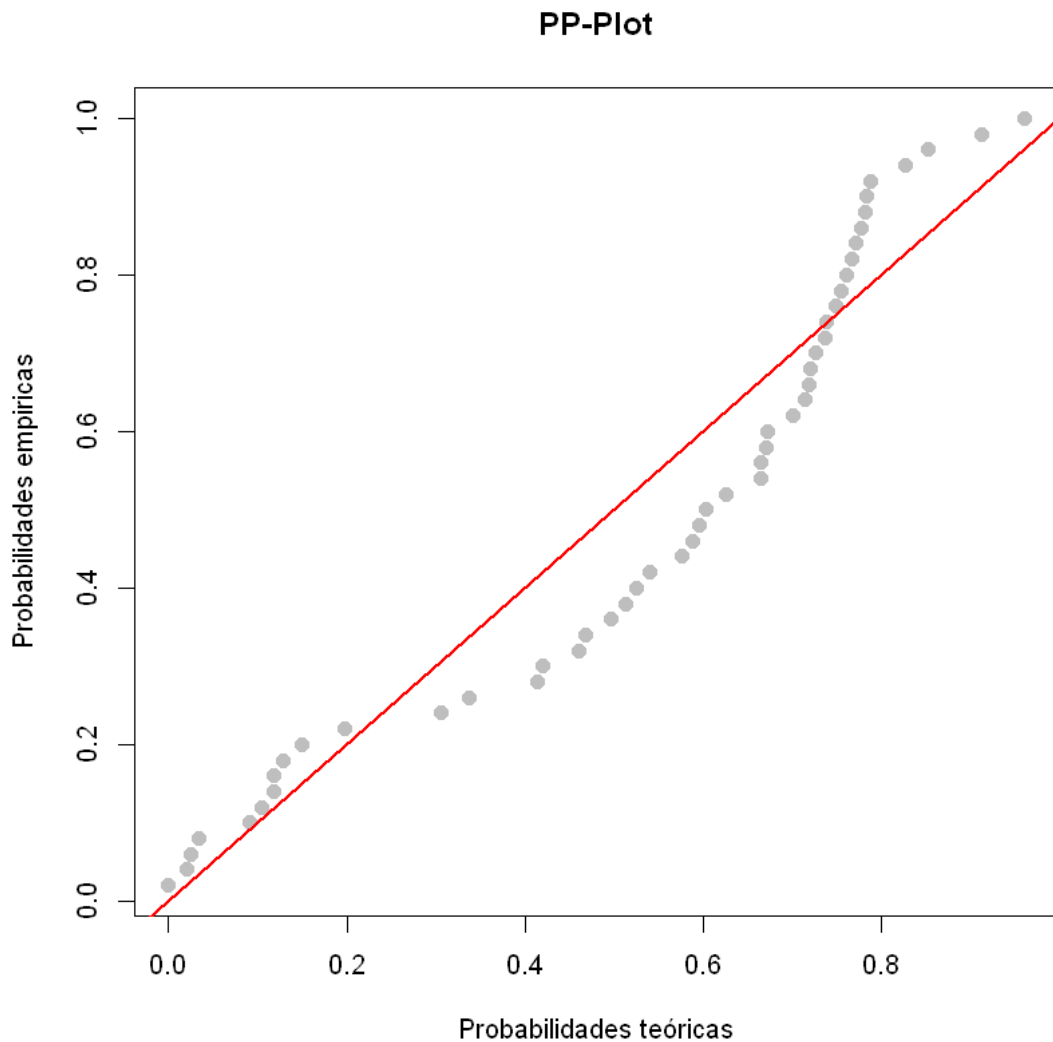
```
[61]: QQplot(x = Y_Residual, distr = "norm", para = PARA_Residual_normal, col =  
→"gray", main = "QQ-Plot", xlab = "Cuantiles teóricos",  
lcol = "red", lwd = 2)
```



El Grafico percentil-percentil (P-P plot).

```
[62]: PPplot(x = Y_Residual, distr = "norm", para = PARA_Residual_normal, col = "gray", main = "PP-Plot", xlab = "Probabilidades teóricas", lcol = "red", lwd = 2)
```





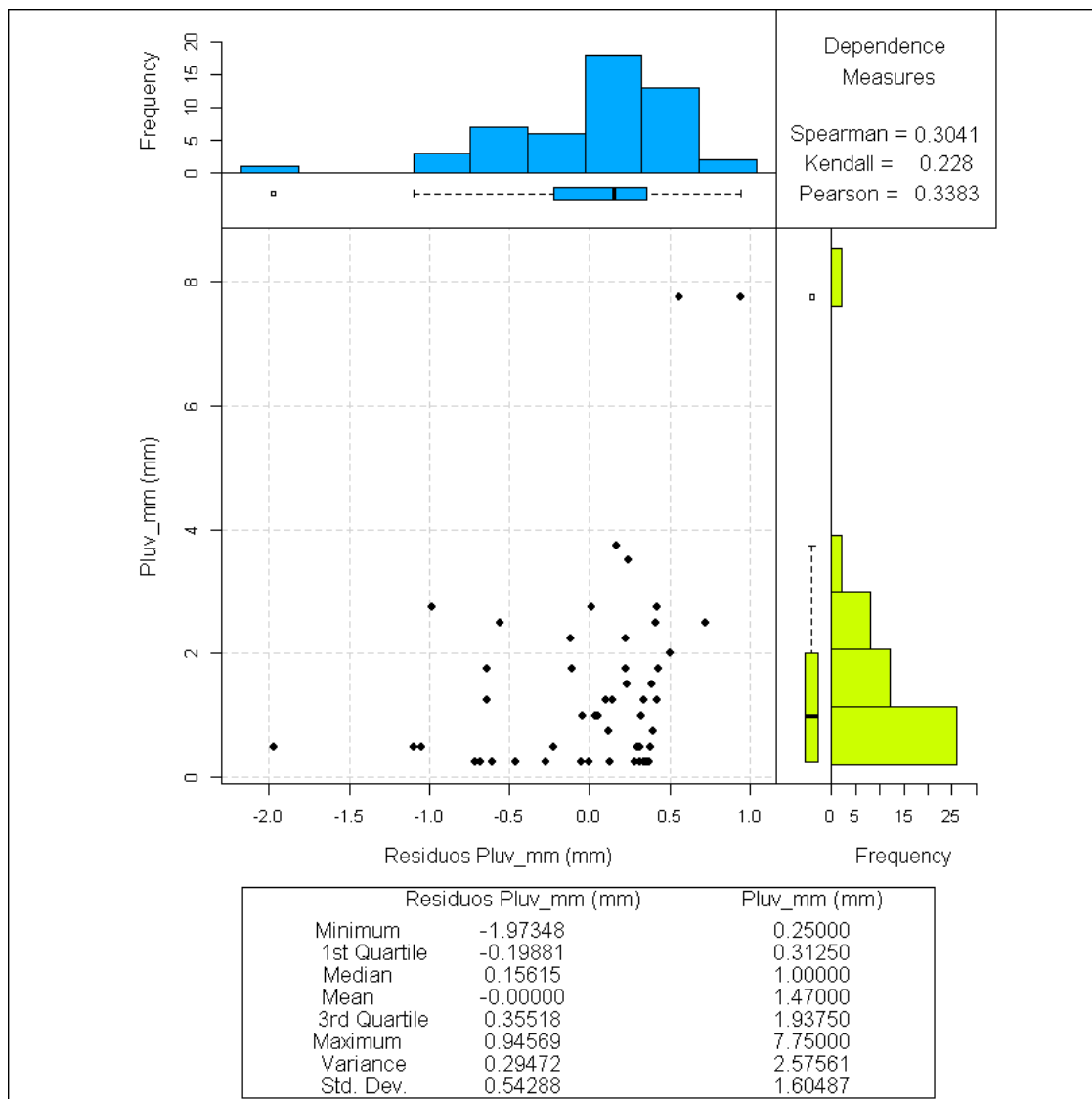
### 3.2.4 Análisis bivariado: Y vs Y residual.

Ahora necesitamos evaluar si los valores obtenidos del pluviómetro tienen una relación con los residuos, para esto necesitamos el gráfico de dispersión. El cual generamos de la siguiente forma:

```
[63]: X<-Y_Residual    # Y_Residual is the independent variable

Y<-Pluv_mm           # Pluv_mm is the dependent variable

ScatterPlot(X, Y, 9,
            Xmin = Y_Residual_Stat[2,2], Xmax = Y_Residual_Stat[7,2],
            Ymin = Pluv_mm_Stat[2,2], Ymax = Pluv_mm_Stat[7,2], XLAB = "Residuos_
            ↪Pluv_mm (mm)", YLAB = "Pluv_mm (mm)")
```



En este grafico podemos notar que la medida de dependencia lineal de Pearson es de 0.339, Spearman es de 0.3041 y Kendall es de 0.228. Esto nos da como conclusión que la regresión lineal no cumple con la condición no correlación entre los residuos y las muestras del pluviómetro.

### 3.3 Análisis estadístico bivariado con variables transformadas.

Ahora haremos el caso donde las variables (Radar\_mm, Pluv\_mm) tienen transformada logarítmica. Se escogió que las variables aleatorias usen esta transformación por los resultados que se tuvieron con la variable aleatoria de las muestras obtenidas con el pluviómetro.

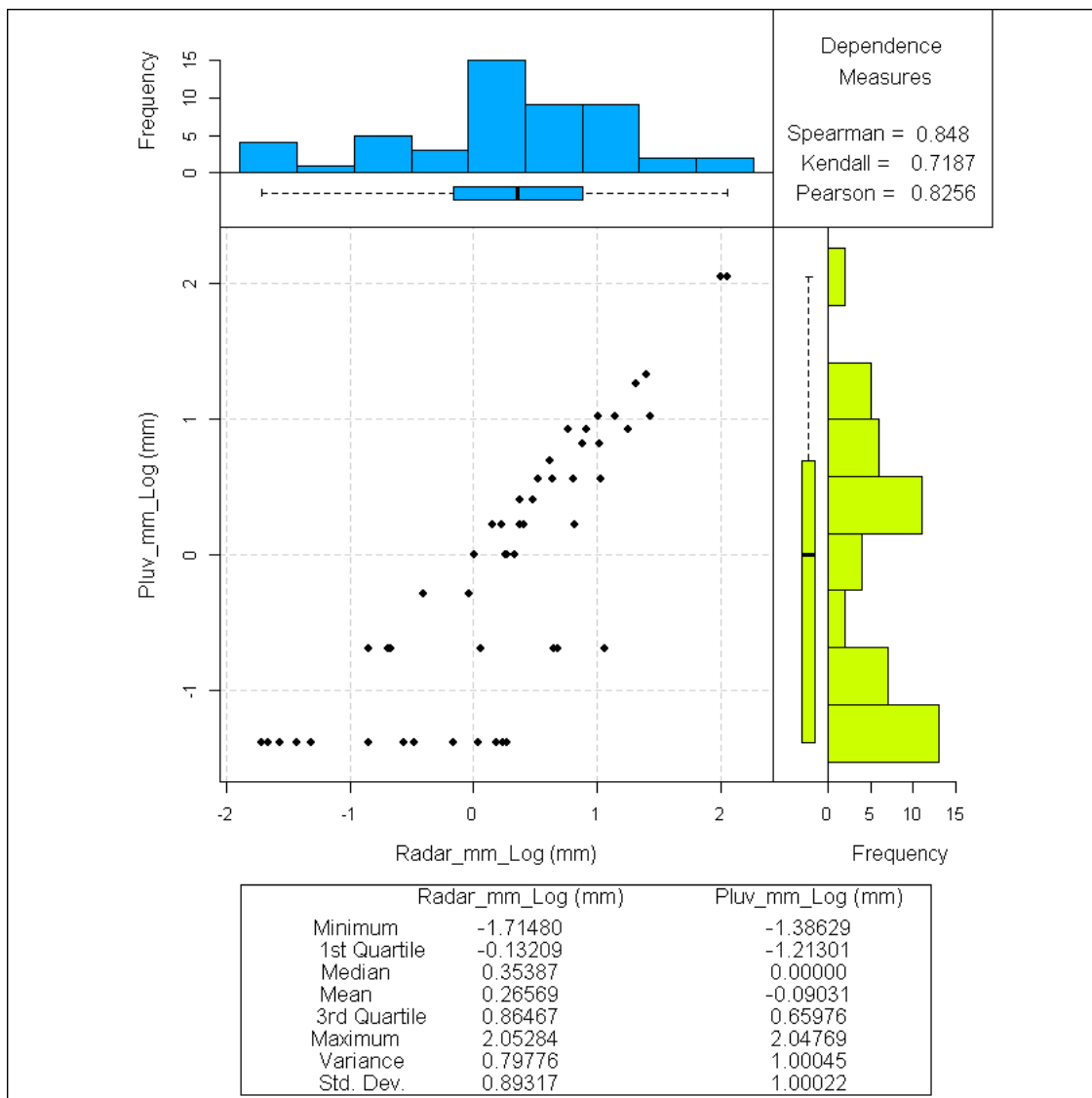
#### 3.3.1 Grafico de dispersión.

En el grafico de dispersión podemos notar que las medidas de dependencia tienen los siguientes valores:

- Spearman (0.848)
- Kendall (0.7187)
- Pearson (0.8256)

Si comparamos las medidas de dependencia de este análisis estadístico bivariado con el anterior análisis podemos notar que el coeficiente de correlación de Pearson cambió de 0.9410 a 0.8256, mientras que los coeficientes de Spearman y Kendall se mantuvieron. esto muestra una gran ventaja al usar los coeficientes de Spearman y Kendall ya que no se ven afectados ante transformaciones.

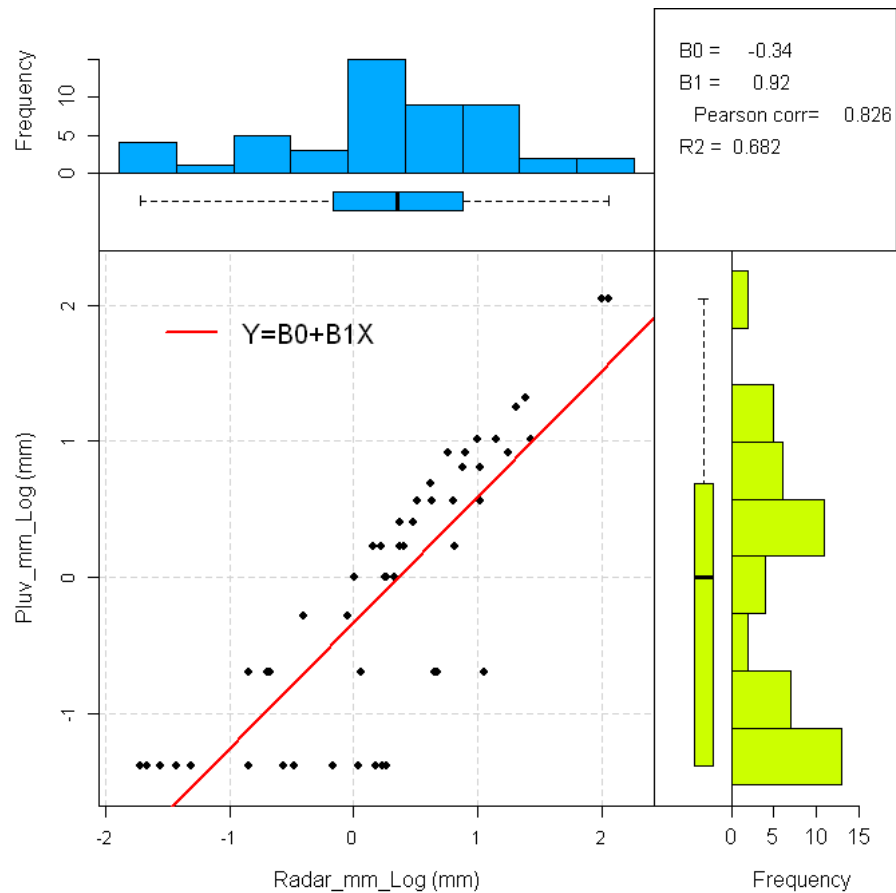
```
[64]: ScatterPlot(Radar_mm_Log , Pluv_mm_Log, 9,
                  Xmin = Radar_mm_Log_Stat[2,2], Xmax = Radar_mm_Log_Stat[7,2],
                  Ymin = Pluv_mm_Log_Stat[2,2], Ymax = Pluv_mm_Log_Stat[7,2],
                  XLAB = "Radar_mm_Log (mm)", YLAB = "Pluv_mm_Log (mm)")
```



### 3.3.2 Analisis de regresión lineal.

Ahora obtendremos los valores de la regresión lineal .

```
[65]: scatterplotReg(Radar_mm_Log , Pluv_mm_Log, 9,  
    Xmin = Radar_mm_Log_Stat[2,2], Xmax = Radar_mm_Log_Stat[7,2],  
    Ymin = Pluv_mm_Log_Stat[2,2], Ymax = Pluv_mm_Log_Stat[7,2],  
    XLAB = "Radar_mm_Log (mm)", YLAB = "Pluv_mm_Log (mm)")
```



los valores de la regresión lineal son:

```
[66]: # Linear Regression  
X<-Radar_mm_Log  
Y<-Pluv_mm_Log
```

```
linear_regression <-lm(Y ~ X)

# Linear Regression Parameters
B0 <- linear_regression$coefficients[1]
B0
B1 <- linear_regression$coefficients[2]
B1
```

**(Intercept):** -0.335971802729909

**X:** 0.924607667074211

Procedemos a calcular los residuos

```
[64]: X<-Radar_mm_Log
      Y<-Pluv_mm_Log

      Y_Regression <- B0 + B1*X
      Y_Residual <- Y-Y_Regression
```

Y calculamos sus estadígrafos e histograma.

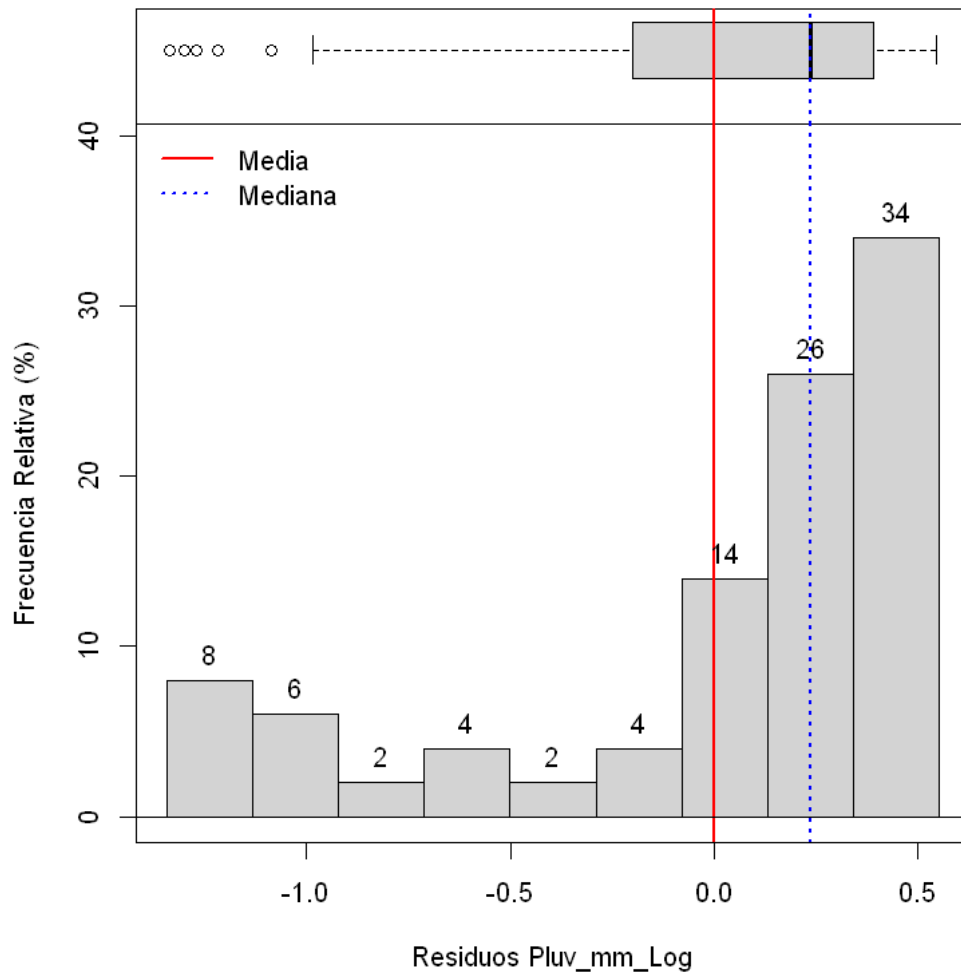
```
[67]: # Regression line and Residual Calculation
      Y_Regression <- linear_regression$fitted.values
      Y_Residual <- linear_regression$residuals

      Y_Residual_Stat<-Estadisticas(Y_Residual)
      write.csv(Y_Residual_Stat , file = "Results/AED/Pluv_mm_Log_Residual_Stat.csv")
      Y_Residual_Stat
```

	Statistics <chr>	Values <dbl>
muestras	n	5.000000e+01
minimos	Minimum	-1.335200e+00
cuantiles1	1st. Quartile	-1.624000e-01
medianas	Median	2.373000e-01
medias	Mean	0.000000e+00
cuantiles3	3rd. Quartile	3.924000e-01
maximos	Maximum	5.445000e-01
rangos	Rank	1.879700e+00
rangosInt	Interquartile Rank	5.548000e-01
varianzas	Variance	3.184000e-01
desvs	Standard Deviation	5.643000e-01
CVs	Variation Coeff.	1.454272e+17
simetrias	Skewness	-1.264700e+00
curtosiss	Kurtosis	3.224800e+00

A data.frame: 14 × 2

```
[68]: HistBoxplot(x=Y_Residual, mean = Y_Residual_Stat[5,2], median = Y_
→Y_Residual_Stat[4,2], main = "",
        xlab = "Residuos Pluv_mm_Log", ylab = "Frecuencia Relativa (%)",
→AbsFreq = FALSE, PercentFreq = TRUE )
```



En este caso podemos ver que el histograma presenta asimetría negativa con cuatro valores atípicos localizados a la izquierda. La diferencia entre la media y la mediana es de 0.2373, la cual es alta.

```
[70]: FitDistr2_Residual_normal<-FitDistribution(data = Y_Residual, DISTR="norm",
→BREAKS = "Sturges", col = "darkgray", DistName = "Normal")
```

```
$start.arg
$start.arg$mean
```

```
[1] 3.88036e-18
```

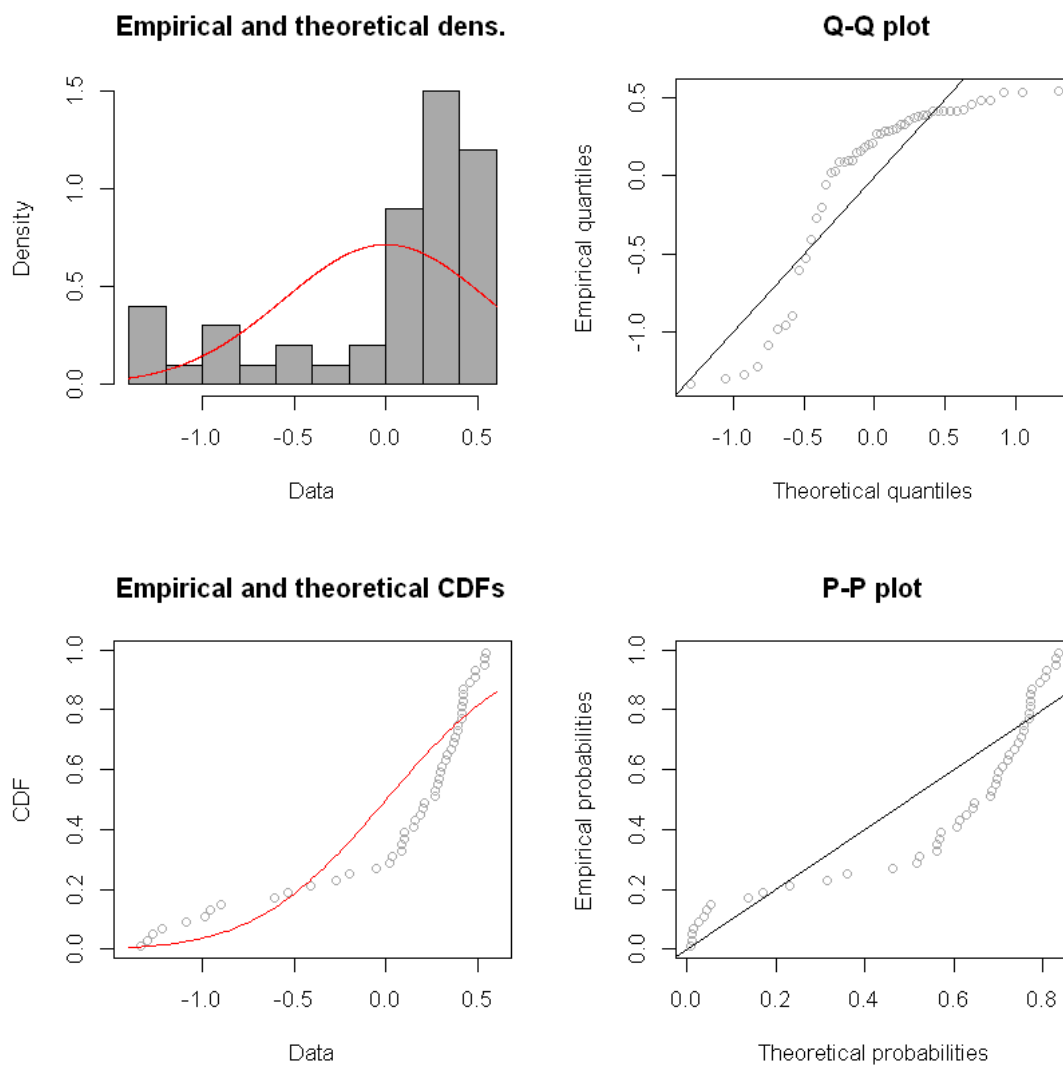
```
$start.arg$sd
```

```
[1] 0.5586384
```

```
$fix.arg
```

```
NULL
```

```
Warning message in hist.default(data, breaks = breaks, plot = FALSE, ...):  
"argument 'col' is not made use of"
```



Analizando los residuos podemos notar que el histograma con la distribución normal (figura su-

perior izquierda) nos muestra que los residuos no son normales, en especial los valores localizados a los extremos del histograma. El grafico Q-Q plot (figura superior derecha) también muestra que pocas muestras están en la recta. La grafica comparativa entre las funciones de distribución acumulativas empírica y teórica (figura inferior izquierda) no muestra un buen ajuste y en el caso del grafico P-P plot (figura inferior derecha) hay muy pocas muestras cercanas a la recta. Con esto podemos concluir que los residuos no cumplen con los requisitos de la regresión lineal.

```
[71]: FD_HT_Residual_normal<-FitDistr2_Residual_normal$x
      FD_HT_Residual_normal
```

	Nombre <chr>	Nivel de significancia <chr>	P-valor <chr>	Estadístico <chr>	Decisión <chr>
A data.frame: 2 × 5	Kolmogorov-Smirnov	0.05	0.004802	0.2412	Rechazo H0
	Anderson-Darling	0.05	0.006424	4.282	Rechazo H0

Las pruebas de normalidad de Kolmogorov-Smirnov y Anderson-Darling confirman lo analizado con los graficos.

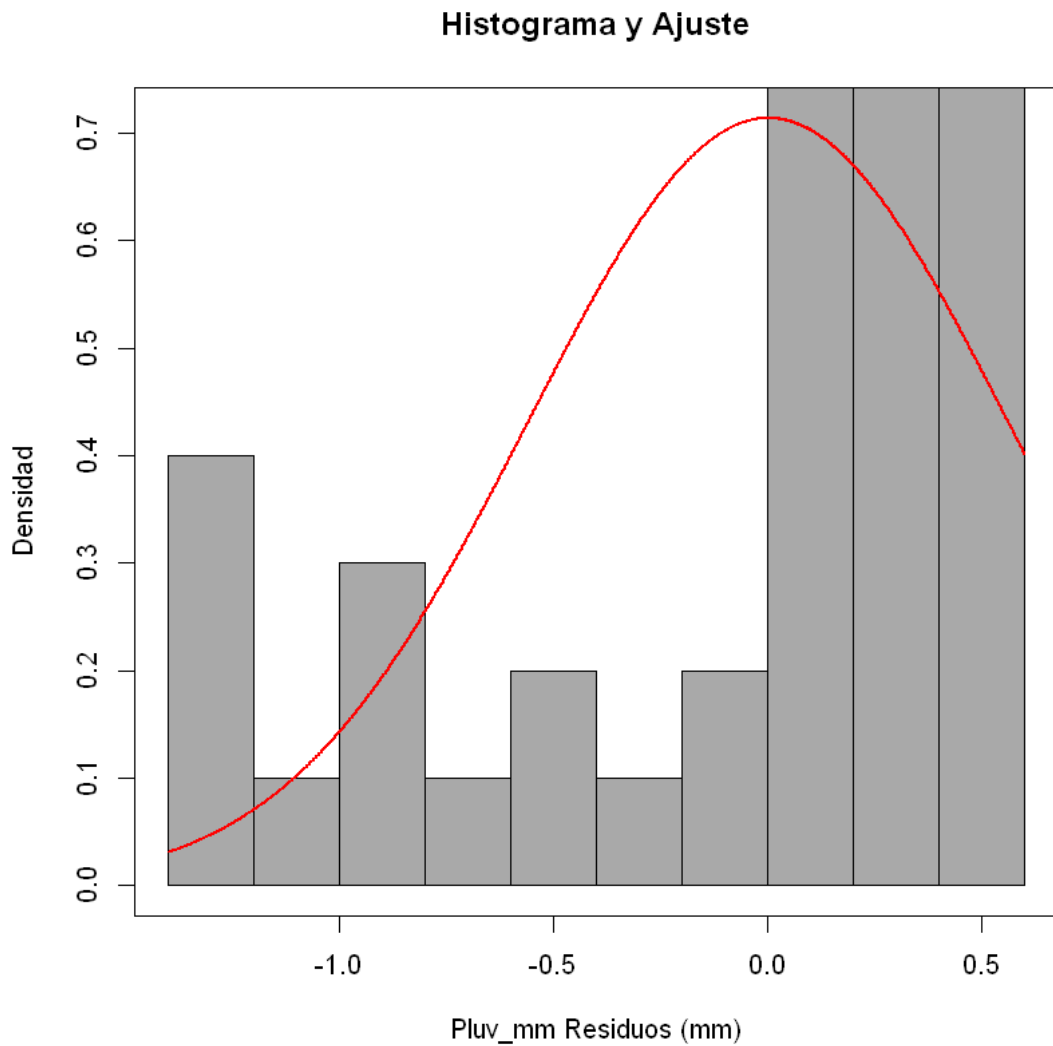
```
[72]: FD_FP_Residual_normal<-FitDistr2_Residual_normal$y
      FD_FP_Residual_normal
```

		Normal <dbl>
A data.frame: 4 × 1	Media	3.880360e-18
	Desviación estándar	5.586384e-01
	Máxima Verosimilitud	-4.183428e+01
	AICC	8.766856e+01

Graficamos las comparativas entre el histograma y la funcion de distribución normal.

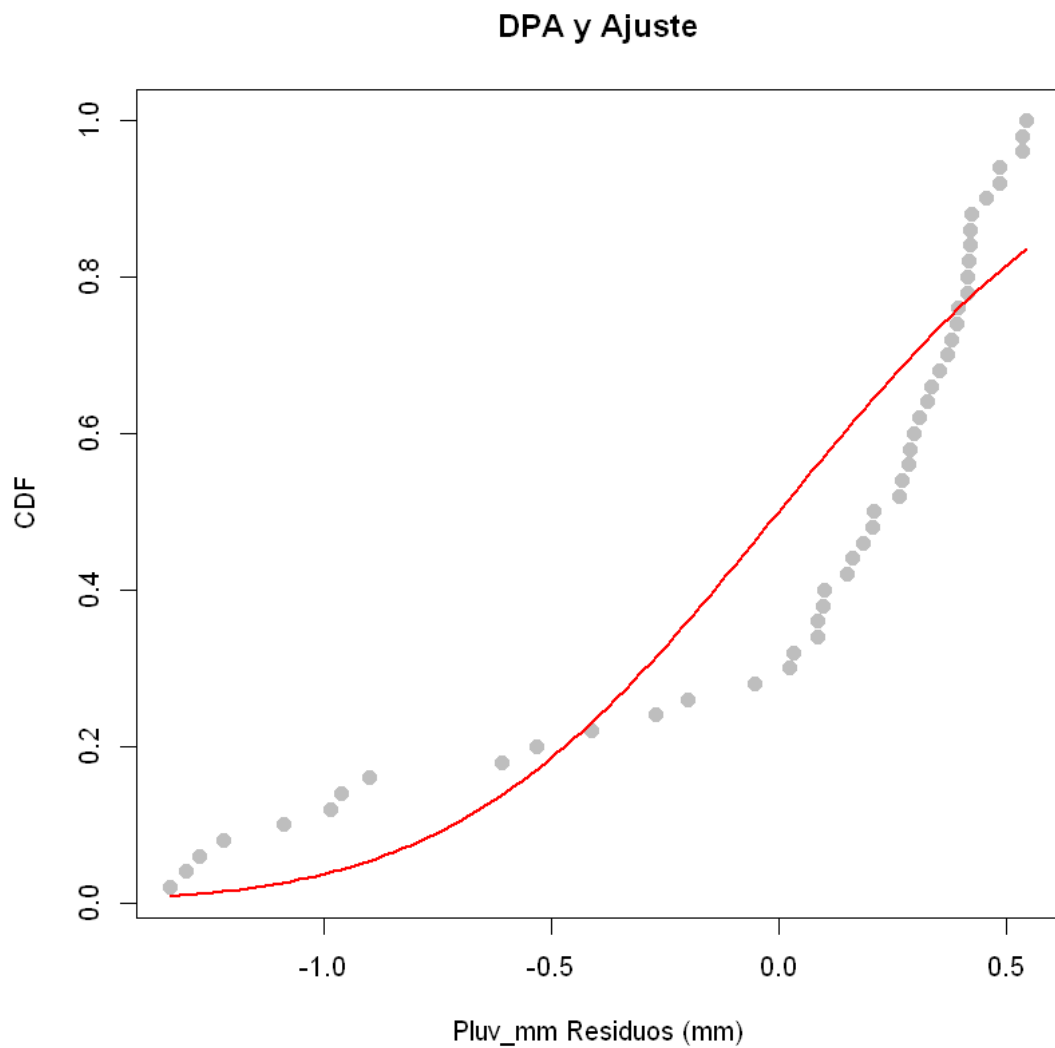
```
[73]: PARA_Residual_normal <- list(mean = as.numeric(FD_FP_Residual_normal[1,1]), sd =
      ↪as.numeric(FD_FP_Residual_normal[2,1]))
      HistModel(x = Y_Residual, distr = "norm", para = PARA_Residual_normal, breaks =
      ↪"Sturges", freq = FALSE, main = "Histograma y Ajuste", xlab = "Pluv_mm Residuos",
      ↪(mm)",
      ylab = "Densidad", colCurve = "red", col = "darkgray")
```





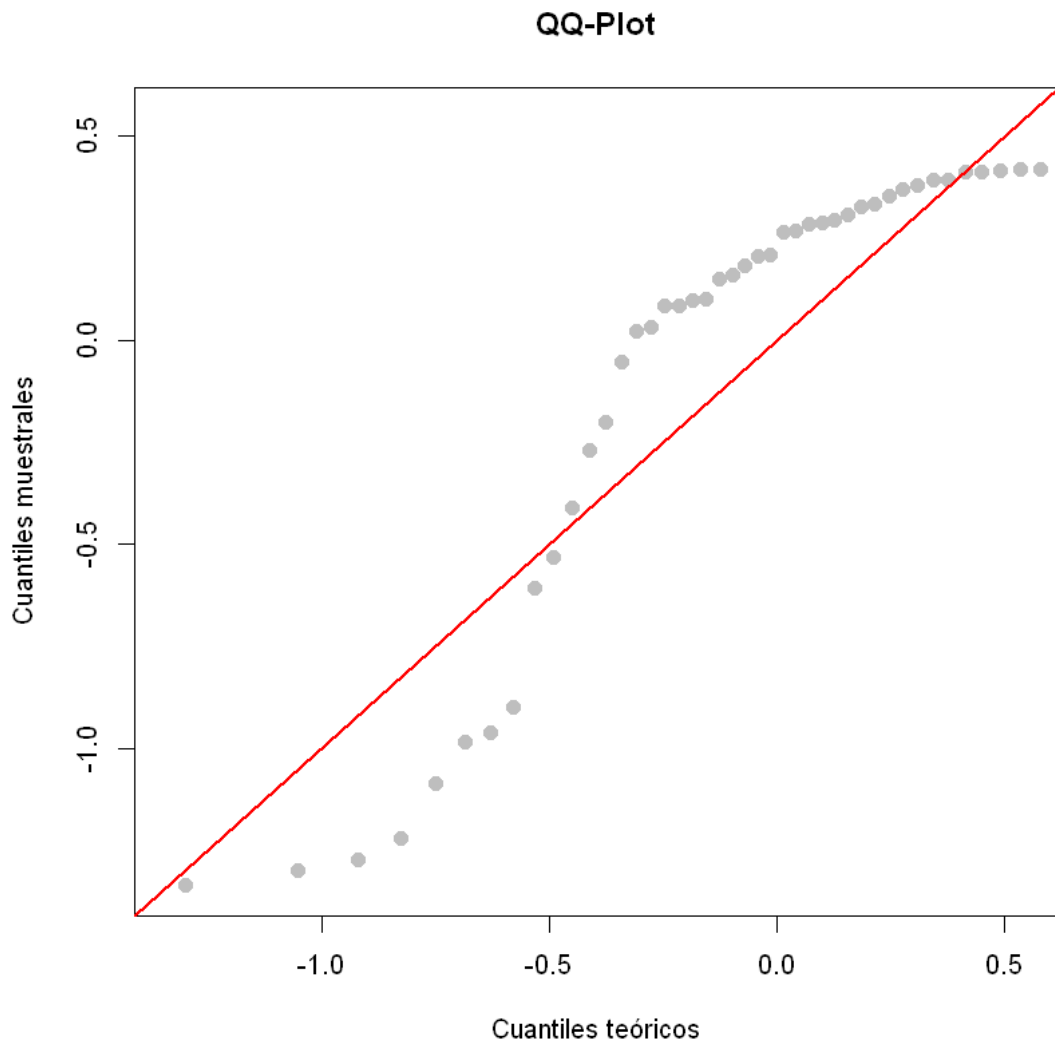
Graficamos las comparativas entre la función de distribución acumulativa empírica y la función de distribución acumulativa normal.

```
[74]: CDF(x = Y_Residual, distr = "norm", para = PARA_Residual_normal, col = "gray",
      ↪main = "DPA y Ajuste", xlab = "Pluv_mm Residuos (mm)",
      lcol = "red", lwd = 2)
```



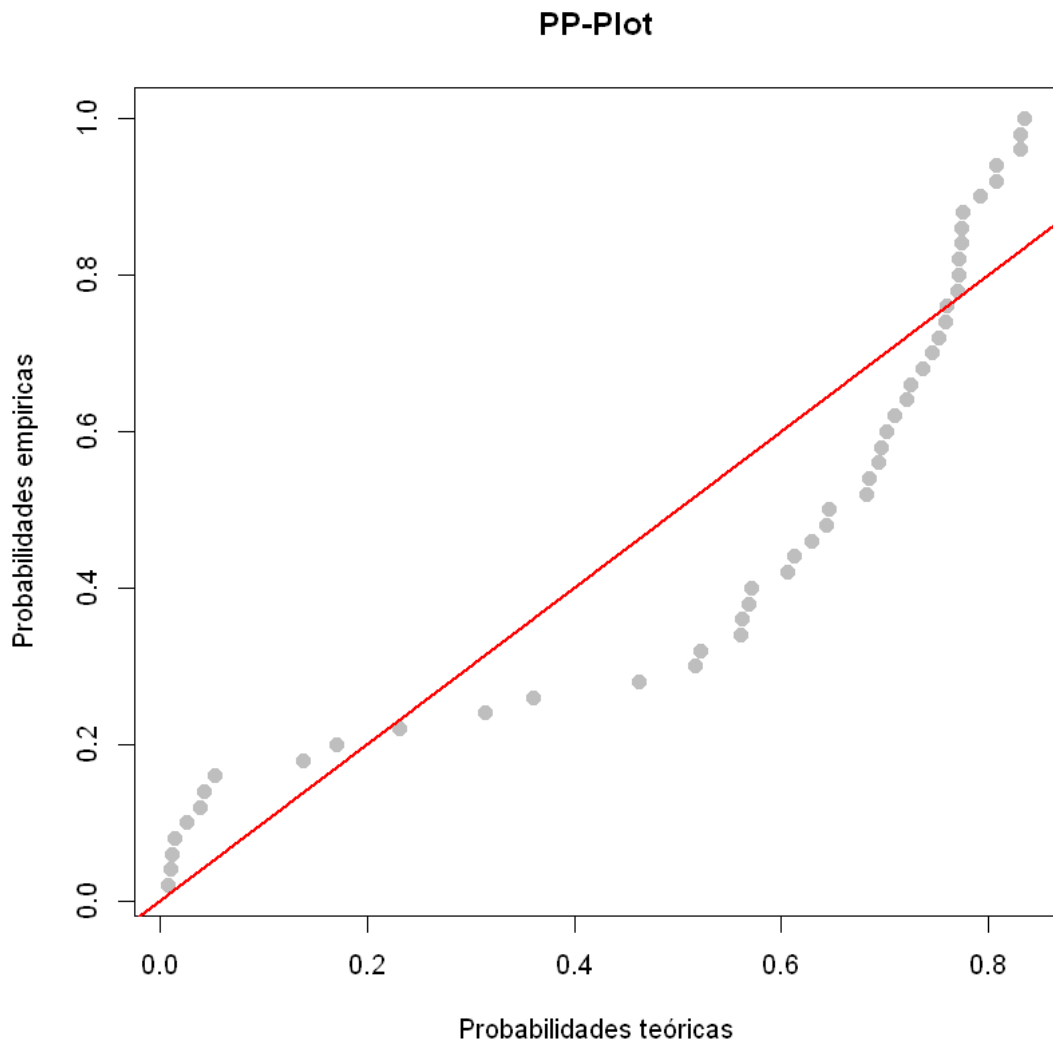
El Grafico cuantil-cuantil (Q-Q plot).

```
[75]: QQplot(x = Y_Residual, distr = "norm", para = PARA_Residual_normal, col = "gray", main = "QQ-Plot", xlab = "Cuantiles teóricos", lcol = "red", lwd = 2)
```



El Grafico Percentil-percentil (P-P plot).

```
[76]: PPplot(x = Y_Residual, distr = "norm", para = PARA_Residual_normal, col = "gray", main = "PP-Plot", xlab = "Probabilidades teóricas", lcol = "red", lwd = 2)
```

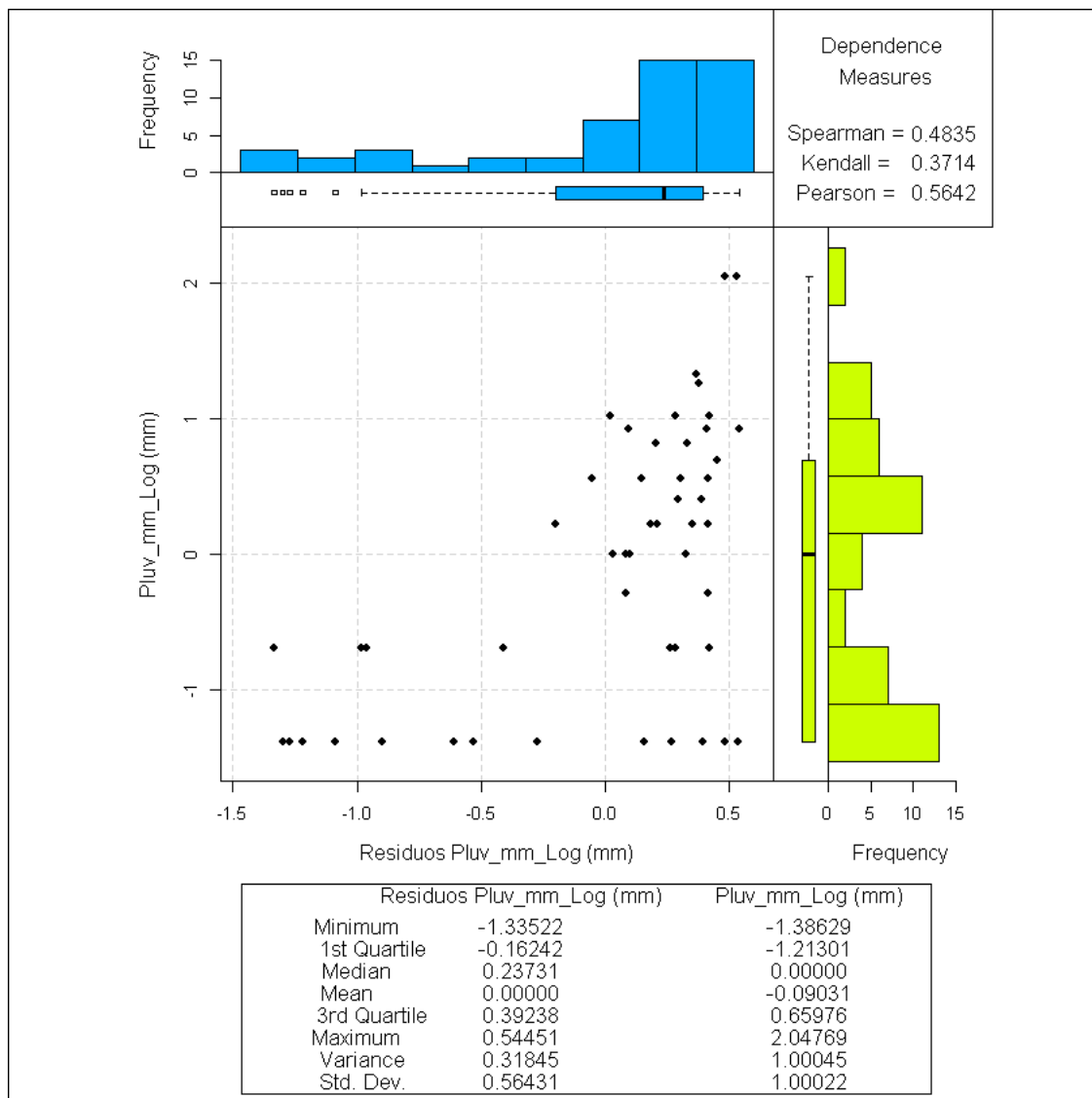


### 3.3.3 Análisis bivariado: Y vs Y residual.

```
[77]: X<-Y_Residual      # Y_Residual es la variable independiente
      Y<-Pluv_mm_Log     # Pluv_mm_Log es la variable dependiente
```

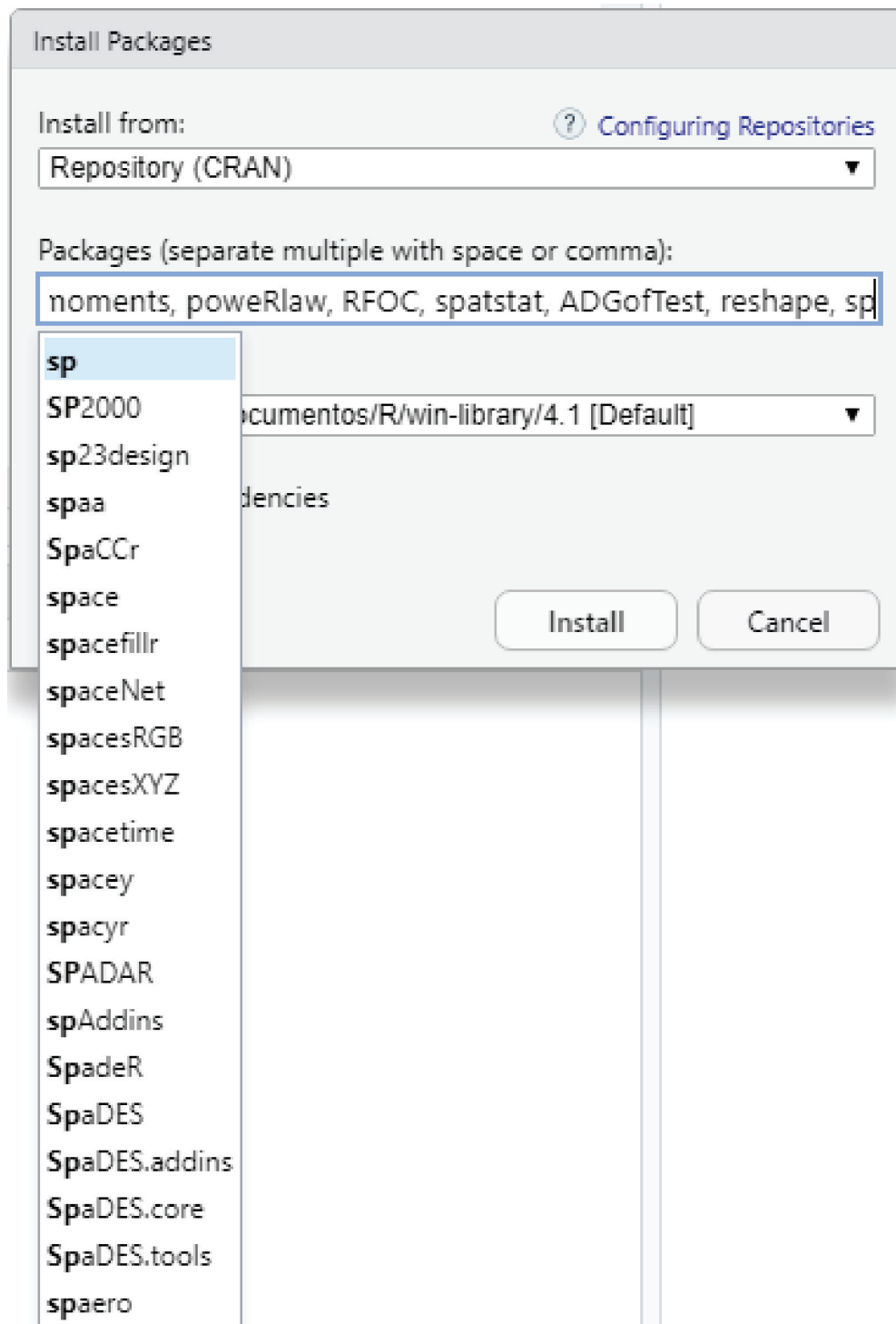
Ahora Analizaremos la dependencia entre la variable aleatoria con transformación logarítmica y sus residuos. Se hace el grafico de dispersión.

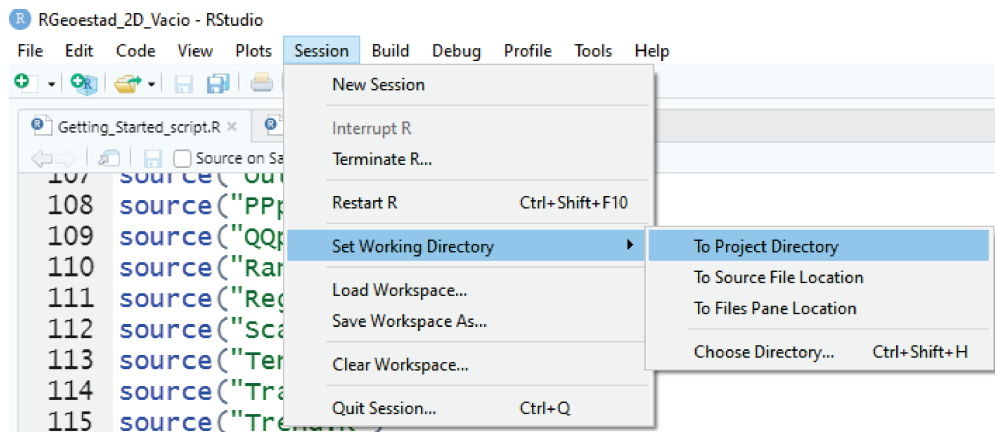
```
[78]: ScatterPlot(X, Y, 9,
                  Xmin = Y_Residual_Stat[2,2], Xmax = Y_Residual_Stat[7,2],
                  Ymin = Pluv_mm_Log_Stat[2,2], Ymax = Pluv_mm_Log_Stat[7,2], XLAB = "
→Residuos Pluv_mm_Log (mm)", YLAB = "Pluv_mm_Log (mm)")
```



El resultado de las medidas de dependencia muestra que la dependencia es considerable. Pearson tiene un valor de 0.5702, Spearman es de 0.4971 y Kendall es de 0.3818. Con esta última prueba podemos concluir que la regresión lineal usando las variables aleatorias transformadas no cumplió con ninguna de las condiciones.

Con el caso de las muestras obtenidas de los pluviómetros, podemos ver que los valores atípicos se localizan en los mismos lugares de los valores atípicos obtenidos del radar meteorológico, por lo que debemos tomar en cuenta esta información para el análisis variográfico.





```

40
41 library(Rcpp)
42 library(maps)
43 library(mapproj)
44 library(actuar)
45 library(fields)
46 library(fitdistrplus)
47 library(geoR)
48 library(gstat)
49 library(MASS)
50 library(moments)
51 library(powerlaw)
52 library(RFOC)
53 library(spatstat)
54 library(ADGofTest)
55 library(reshape)
56 library(sp)

```

