R_Basico_2

J.Ballesta

10/12/25

Resumen

En esta publicación, continuaremos con el uso de R a nivel básico y presentaremos el paquete ggplot2, mostrando alguna de sus capacidades para generar gráficos sobre nuestros datos de interés.

In this publication, we will continue with the basic use of R and introduce the ggplot2 package, showcasing some of its capabilities for generating graphs based on our data of interest.

In diesem Beitrag setzen wir unsere Auseinandersetzung mit R auf grundlegendem Niveau fort und stellen das Paket ggplot2 vor. Wir werden einige seiner Funktionen zur Erstellung von Grafiken auf Basis unserer interessierenden Daten demonstrieren.

Quarto : YAML

El bloque de código en la cabecera del script cuando seleccionamos salida para Quarto en RStudio (RStudio Team 2020), recibe el nombre de YAML, es donde se configuran todas las opciones de formato de salida de Quarto :

- HTML
- PDF
- MS Word
- Markdown
- Revealjs
- PowerPoint
- …

En este caso, hemos optado por usar un *format* para salida por documento (tipo .docx), que después podremos editar en caso de que sea necesario. Para ver el YAML es necesario inspeccionar el código fuente del archivo de salida.

Introducción

Seguimos trabajando con R(R Core Team 2024), Quarto y RStudio(RStudio Team 2020), para conocer las posibilidades de análisis de datos que nos ofrecen. En esta ocasión haremos uso del package *ggplot2* (Wickham 2016a), para los gráficos de los datos.

```
#
#    En este análisis vamos a trabajar con el dataset faithful disponible en R, que contiene
# el tiempo en minutos de la duración de la erupción y el tiempo en minutos entre erupciones
# del Old Faithful geyser en Yellowstone National Park.
```

```
help (faithful)
# como siempre cargamos los datos en una variable local de trabajo
datos <- faithful
# comprobamos los 5 primeros y últimos valores
head(datos, 5)
```

```
  eruptions waiting
1     3.600      79
2     1.800      54
3     3.333      74
4     2.283      62
5     4.533      85
```

```
#
tail(datos, 5)
```

```
    eruptions waiting
268     4.117      81
269     2.150      46
270     4.417      90
271     1.817      46
272     4.467      74
```

```
# comprobamos el nombre de los campos de datos en la variable de trabajo
names(datos)
```

```
[1] "eruptions" "waiting"
```

Análisis preliminar de los datos

En este caso el tipo de datos es un *data.frame y* está compuesto por dos campos.

```
#
#    Vemos la estructura de los datos, para conocer el número de observaciones, tipo de datos y las variables que incluye.
str(datos)
```

```
'data.frame':   272 obs. of  2 variables:
 $ eruptions: num  3.6 1.8 3.33 2.28 4.53 ...
 $ waiting  : num  79 54 74 62 85 55 88 85 51 85 ...
```

```
# hacemos un resumen de algunas magnitudes estadísticas del conjunto de datos (en este caso una
# posible alternativa a la función summary() es la función fivenum())
summary(datos)
```

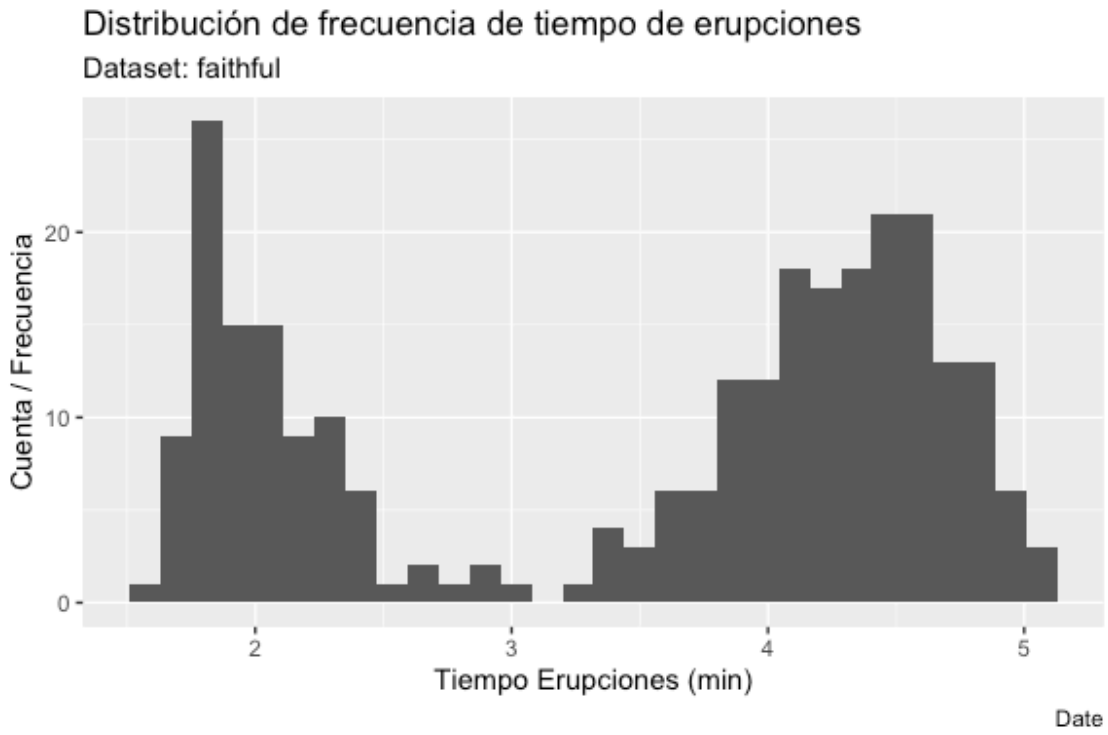
```
   eruptions        waiting
 Min.   :1.600   Min.   :43.0
 1st Qu.:2.163   1st Qu.:58.0
 Median :4.000   Median :76.0
 Mean   :3.488   Mean   :70.9
 3rd Qu.:4.454   3rd Qu.:82.0
 Max.   :5.100   Max.   :96.0
```

Análisis gráfico

Para esta publicación usaremos *ggplot2*(Wickham 2016a), uno de los paquetes en R (R Core Team 2024) para la preparación de gráficos con más difusión y apoyo de la comunidad .

```
#
#    El primer paso es cargar la librería de gráficos que vamos a usar en nuestro estudio.
library(ggplot2)
# este package nos dará las funciones necesarias para formatear los valores de etiqueta en
#los ejes.
library(scales)
#
# ggplot2 trabaja por capas (geom) que van componiendo la salida gráfica que deseamos para
# nuestro estudio. En este caso, presentaremos nuestros datos desde la capa general y
después
# añadiendo una capa adicional para el gráfico de histograma de cada variable.
ggplot(data=datos, aes(x= eruptions))+
    geom_histogram()+
    labs ( title="Distribución de frecuencia de tiempo de erupciones",
           subtitle ="Dataset: faithful",
           caption ="Date")+
    xlab ("Tiempo Erupciones (min)")+
    ylab ("Cuenta / Frecuencia")
```

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

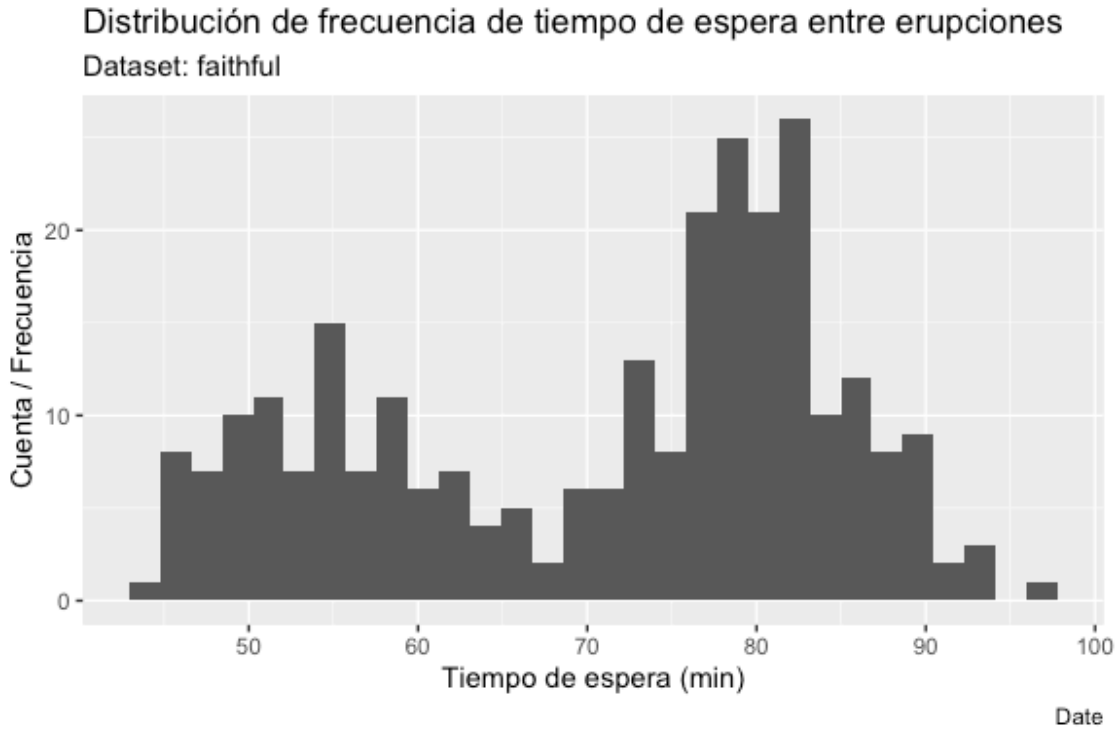



```
#

#
#
```

```
ggplot(data=datos, aes(x= waiting))+
    geom_histogram()+
    labs ( title="Distribución de frecuencia de tiempo de espera entre erupciones",
           subtitle ="Dataset: faithful",
           caption ="Date")+
    xlab ("Tiempo de espera (min)")+
    ylab ("Cuenta / Frecuencia")
```

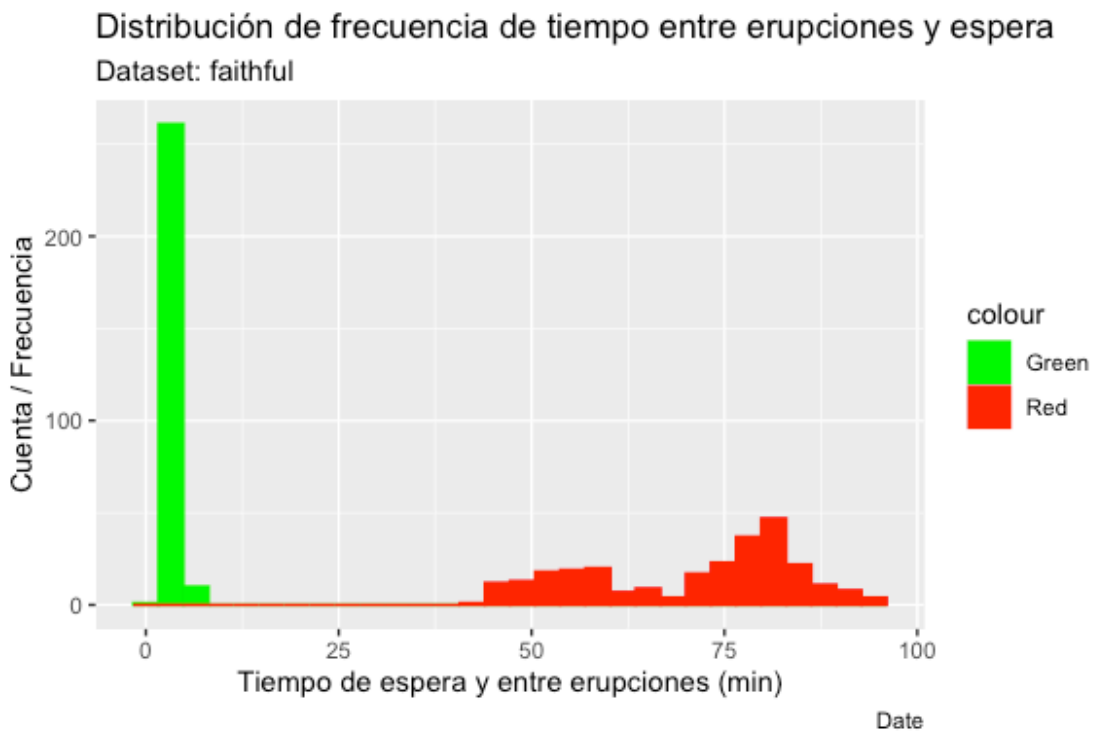
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.



En los histogramas podemos ver que la distribución de ambos tiempos parece seguir un tipo de distribución bimodal. En algunos casos, aprovechando las capacidades de *ggplot2* (Wickham 2016b) nos interesará superponer ambos histogramas. Vemos a continuación como:

```
#
#    superponemos ambos histogramas de tiempo de erupciones (en verde) y tiempo de espera
# entre erupciones (en rojo) y para ello cada grupo de datos irá en su propia capa de
# geom_histogram()
ggplot(data=datos) +
    geom_histogram(aes(x=eruptions, color="Green"), fill="Green")+
    geom_histogram(aes(x=waiting, color="Red"), fill="Red")+
    scale_color_manual(values= c("Green"="green", "Red"="red"))+
    labs ( title="Distribución de frecuencia de tiempo entre erupciones y espera",
           subtitle ="Dataset: faithful",
           caption ="Date")+
    xlab ("Tiempo de espera y entre erupciones (min)")+
    ylab ("Cuenta / Frecuencia")
```

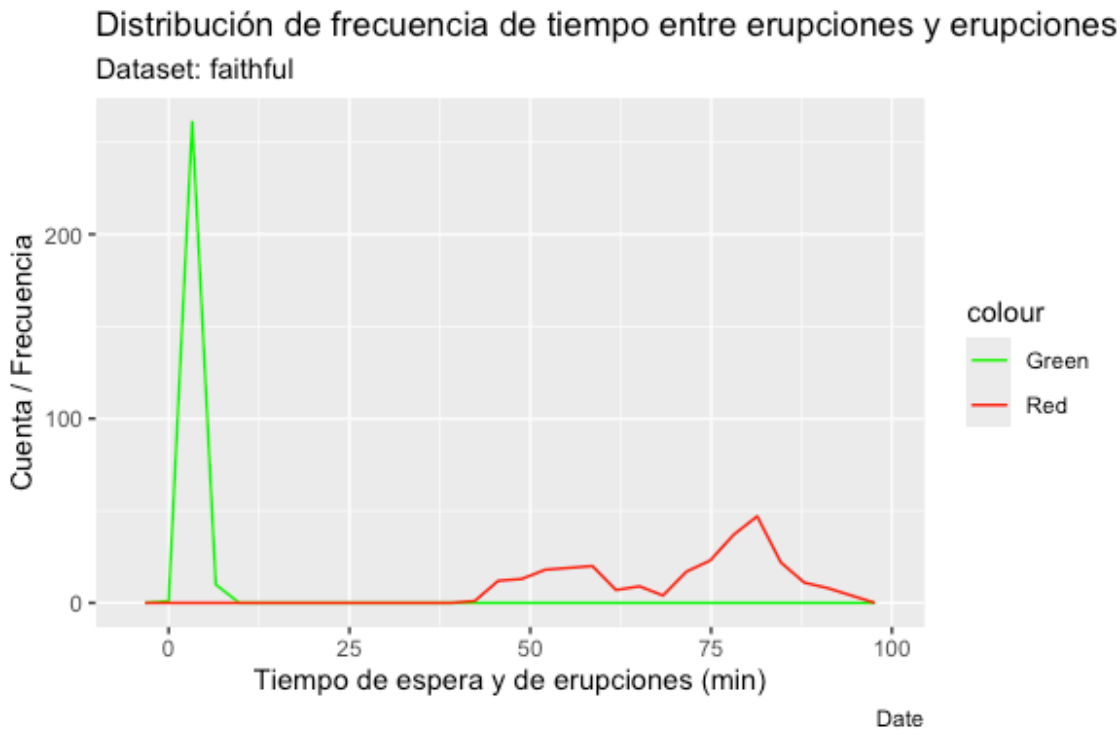
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



En este caso vemos que los datos son difíciles de visualizar con un histograma superponiendo los datos, *ggplot2* (Wickham 2016b) nos proporciona la capa de polígono de frecuencias (*geom_freqpoly()*) para dar otra vista a los datos, en este caso el gráfico quedaría :

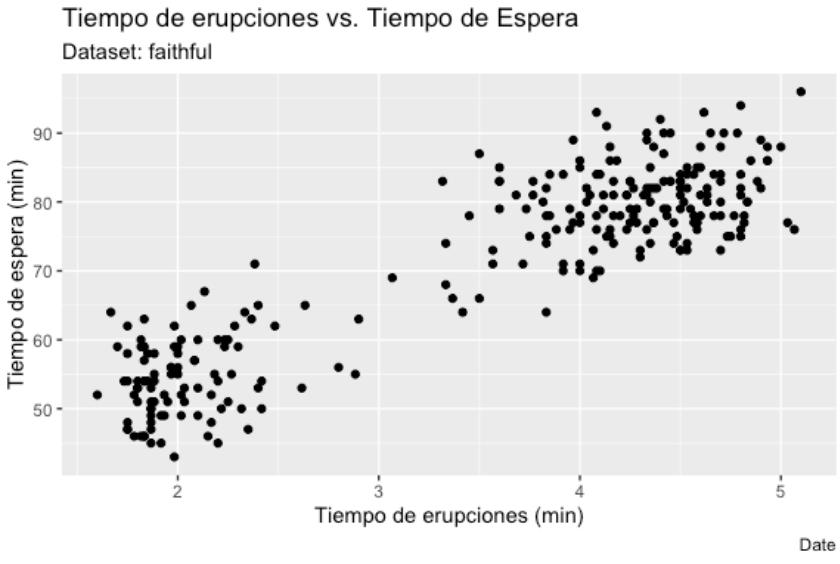
```
#
#    presentamos el tiempo de erupciones y espera entre erupciones como un polígono de
# frecuencias para mejorar la legibilidad de los datos.
ggplot(data=datos)+
    geom_freqpoly(aes(x=eruptions, color="Green"))+
    geom_freqpoly(aes(x=waiting, color="Red"))+
    scale_color_manual(values= c("Green"="green", "Red"="red"))+
    labs (title="Distribución de frecuencia de tiempo entre erupciones y erupciones",
          subtitle ="Dataset: faithful",
          caption ="Date")+
    xlab ("Tiempo de espera y de erupciones (min)")+
    ylab ("Cuenta / Frecuencia")

`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



En este caso, que disponemos de un conjunto de dos variables podemos optar por un gráfico XY, y representar con la capa *geom_point()*, como ambas variables se relacionan entre si :

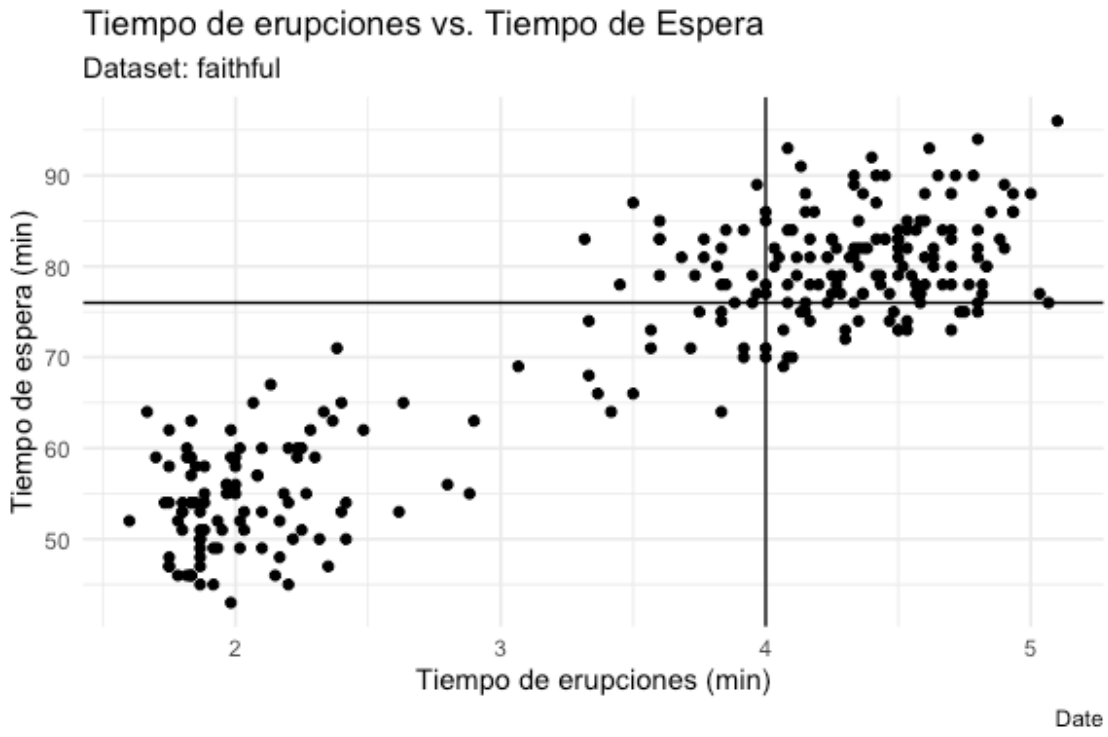
```
#
#    representamos ambas variables de la variable datos en un gráfico XY
ggplot(data=datos, aes(x=eruptions, y=waiting))+
    geom_point()+
    labs (title="Tiempo de erupciones vs. Tiempo de Espera",
          subtitle ="Dataset: faithful",
          caption ="Date")+
    xlab ("Tiempo de erupciones (min)")+
    ylab ("Tiempo de espera (min)")
```



Del gráfico Figure 1 vemos que parece haber una correlación positiva entre ambos tiempos, y una cierta ordenación en dos grupos de los valores.

Para tener algún tipo de referencia añadiremos un par de líneas con la mediana de ambos conjuntos de datos, con las capas *geom_hline()* y *geom_vline()*.

```
#
#    añadimos al gráfico la posición de los valores medianos de ambas variables
ggplot(data= datos, aes(x= eruptions, y= waiting))+
    geom_point()+
    geom_hline(aes(yintercept= median(waiting)))+
    geom_vline(aes(xintercept= median(eruptions)))+
    labs ( title= "Tiempo de erupciones vs. Tiempo de Espera",
           subtitle= "Dataset: faithful",
           caption= "Date")+
    xlab ("Tiempo de erupciones (min)")+
    ylab ("Tiempo de espera (min)")+
    theme_minimal()     # en este caso aplicaremos al gráfico el tema minimal.
```

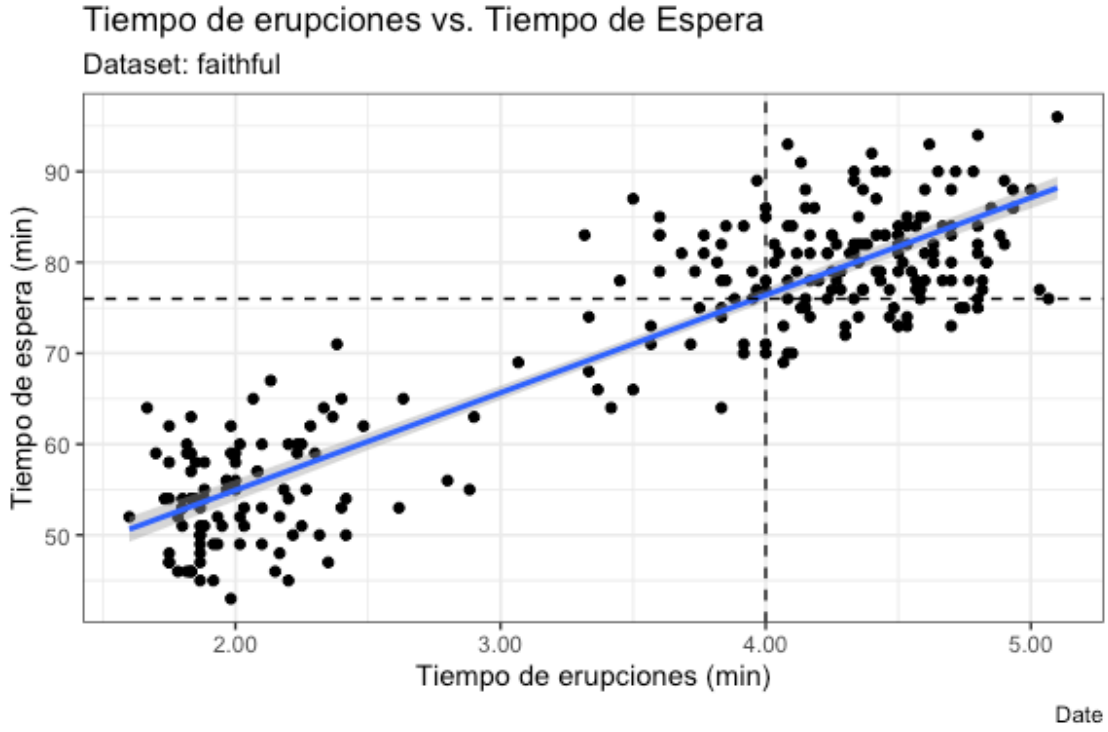


Finalmente para terminar con esta publicación, la parte de análisis gráfico, *ggplot2* (Wickham 2016c) nos proporciona la capa *geom_smooth()* que interpola una curva en los valores del gráfico, en este caso optamos por una regresión lineal, entre los datos disponibles.

```
#
#    en este ejemplo cambiamos el tipo de línea de referencia del valor mediano y añadimos
una # interpolación lineal entre ambos valores, mostrando el error standard (si no queremos
mostrar el error standard en la capa de geom_smooth() incluir se=FALSE)
ggplot(data= datos, aes(x= eruptions, y= waiting))+
    geom_point()+
```

```
    geom_smooth(method= lm)+
    geom_hline(aes(yintercept= median(waiting)), linetype= "dashed")+
    geom_vline(aes(xintercept= median(eruptions)), linetype="dashed")+
    scale_x_continuous(labels = label_number(accuracy = 0.01))+ # formateamos las etiquetas
en el eje X para que aparezcan con dos decimales
    labs (title= "Tiempo de erupciones vs. Tiempo de Espera",
          subtitle= "Dataset: faithful",
          caption= "Date")+
    xlab ("Tiempo de erupciones (min)")+
    ylab ("Tiempo de espera (min)")+
    theme_bw()    # elegimos un tema en blanco y negro para el gráfico
```

`geom_smooth()` using formula = 'y ~ x'



Histograma : Número de Bin.

Existen distintos criterios para establecer el número de bins (clases) en un histograma :

- La regla de Sturges.
- La regla de Scott
- La regla de Freedman-Diaconis
- Prueba empírica (prueba y error)

veremos a continuación un ejemplo con cada uno de ellos.

```
#
ggplot(data=datos, aes(x= eruptions))+
    geom_histogram( binwidth = log2(nrow(datos)+1))+ # Regla de Sturges
```

```
  labs ( title="Distribución de frecuencia de tiempo de erupciones",
         subtitle ="Dataset: faithful",
         caption ="Date")+
  xlab ("Tiempo de erupciones (min)")+
  ylab ("Cuenta / Frecuencia")+
  theme_bw()
```

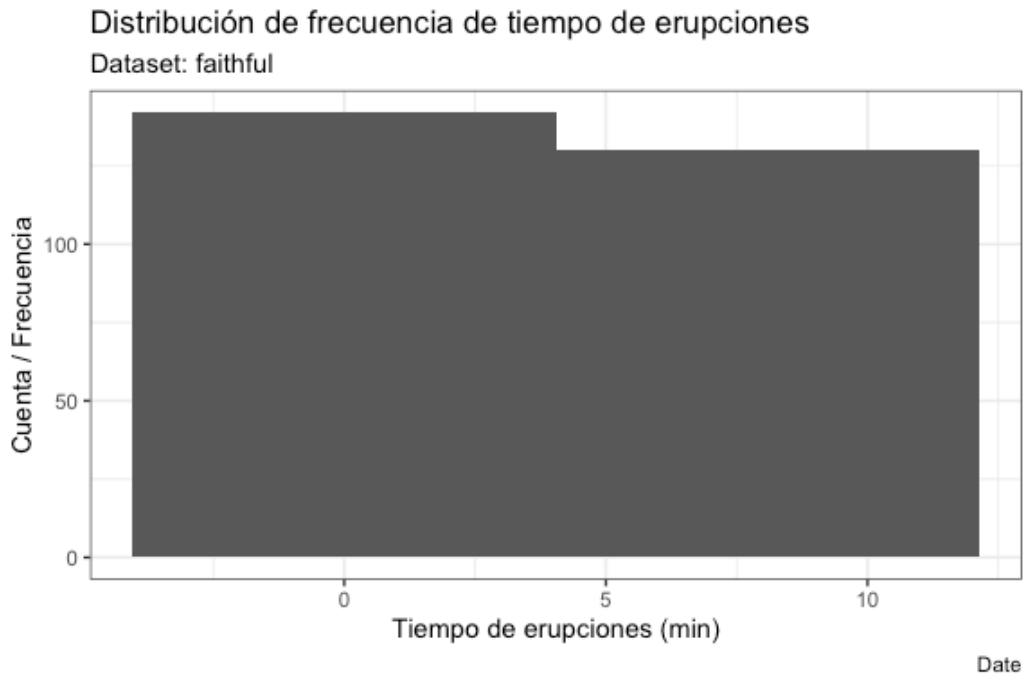
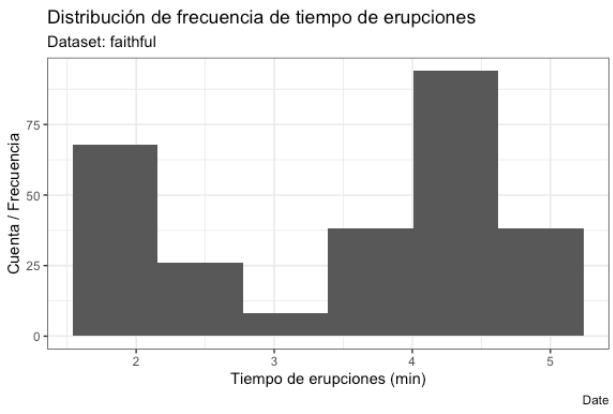


Figure 2: Distribución criterio Sturges

```
#
ggplot(data=datos, aes(x= eruptions))+
    geom_histogram(binwidth=(3.5*sd(datos$eruptions))/nrow(datos)^(1/3))+ # Aplicamos la
regla de Scott
    labs ( title="Distribución de frecuencia de tiempo de erupciones",
           subtitle ="Dataset: faithful",
           caption ="Date")+
    xlab ("Tiempo de erupciones (min)")+
    ylab ("Cuenta / Frecuencia")+
    theme_bw()
```



```
#
ggplot(data=datos, aes(x= eruptions))+
    geom_histogram(binwidth=(2*IQR(datos$eruptions))/nrow(datos)^(1/3))+ # Aplicamos la
regla de Friedman-Diaconis
    labs ( title="Distribución de frecuencia de tiempo de erupciones",
           subtitle ="Dataset: faithful",
           caption ="Date")+
    xlab ("Tiempo de Erupciones (min)")+
    ylab ("Cuenta / Frecuencia")+
    theme_bw()
```

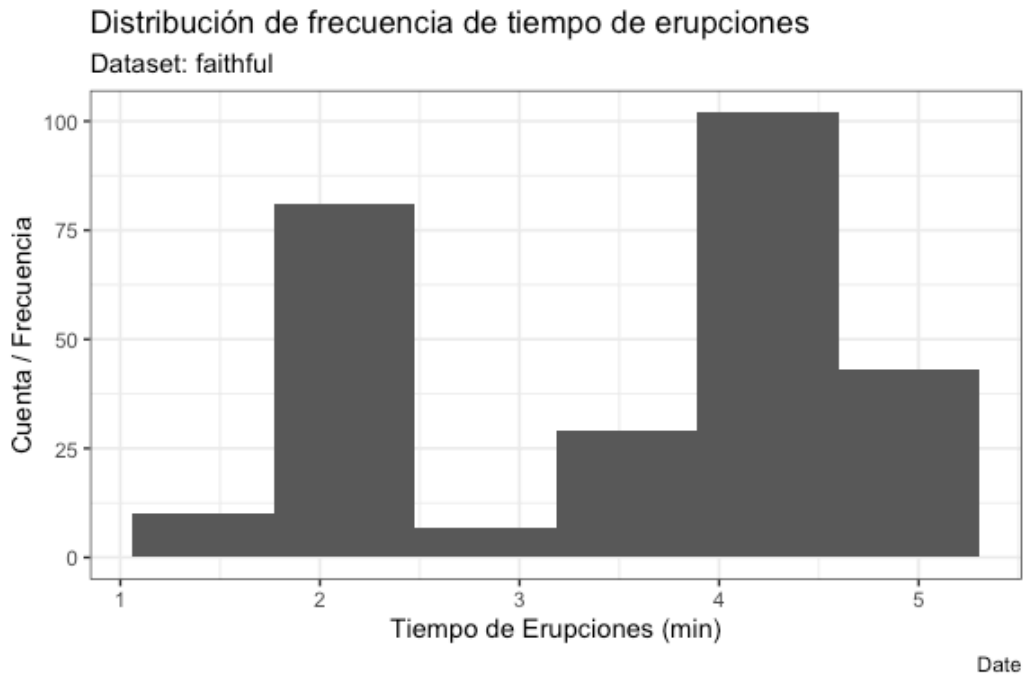


Figure 4: Distribución criterio Friedman-Diaconis

Podemos ver de los gráficos anteriores Figure 2 , Figure 3 y Figure 4 como el contorno del histograma se modifica en función de la anchura del bin elegido.

Referencias

R Core Team. 2024. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

RStudio Team. 2020. *RStudio: Integrated Development Environment for r*. Boston, MA: RStudio, PBC. http://www.rstudio.com/.

Wickham, Hadley. 2016a. “Ggplot2: Elegant Graphics for Data Analysis.” https://ggplot2.tidyverse.org.

———. 2016b. “Ggplot2: Elegant Graphics for Data Analysis.” https://ggplot2.tidyverse.org.

———. 2016c. “Ggplot2: Elegant Graphics for Data Analysis.” https://ggplot2.tidyverse.org.