

ANÁLISIS Y TRATAMIENTO DE DATOS CON R

Con ejemplos e ilustraciones

Primera Edición

Diego Paul Huaraca S.
MS-PLUS, INC.

Un aporte de Source Stat Lab Ecuador a la sociedad.

Índice general

1. Entornos de desarrollo	3
1.1. RStudio	3
1.1.1. Instalación y actualización	4
1.1.2. Funcionamiento	5
1.1.3. Proyectos	6
1.1.4. Ancho de impresión	8
1.1.5. Prompt	9
1.1.6. Decimales	9
1.1.7. Respalando información	10
1.2. R Analytic Flow	11
1.2.1. Ventajas	11
1.2.2. Desventajas	11
1.2.3. Funcionamiento	12

1

Entornos de desarrollo

Un entorno de desarrollo integrado, también conocido como IDE (Integrated Development Environment) es un programa informático compuesto por un conjunto de herramientas de programación que contiene: un editor, un compilador, un depurador y un constructor de interfaz gráfica, el mismo que viene empaquetado como una **aplicación** que facilita de sobremanera la realización de operaciones al usuario mediante una serie de menús o mediante interacción con los objetos gráficos que aparecen en pantalla, a través de periféricos como: el ratón y teclado.

En este capítulo analizaremos dos IDE's de gran importancia y utilidad para los usuarios de R:

- RStudio
- R Analytic Flow

1.1. RStudio

RStudio es un entorno IDE de código abierto lanzado en Febrero 2011 el cual incluye una consola, un editor resaltado de texto que admite la ejecución directa, así como herramientas para gráficos, historial, depuración y gestión del espacio de trabajo. RStudio se encuentra disponible para todas las plataformas (Windows, Mac, Linux), además puede ser ejecutado a través de un navegador web¹.

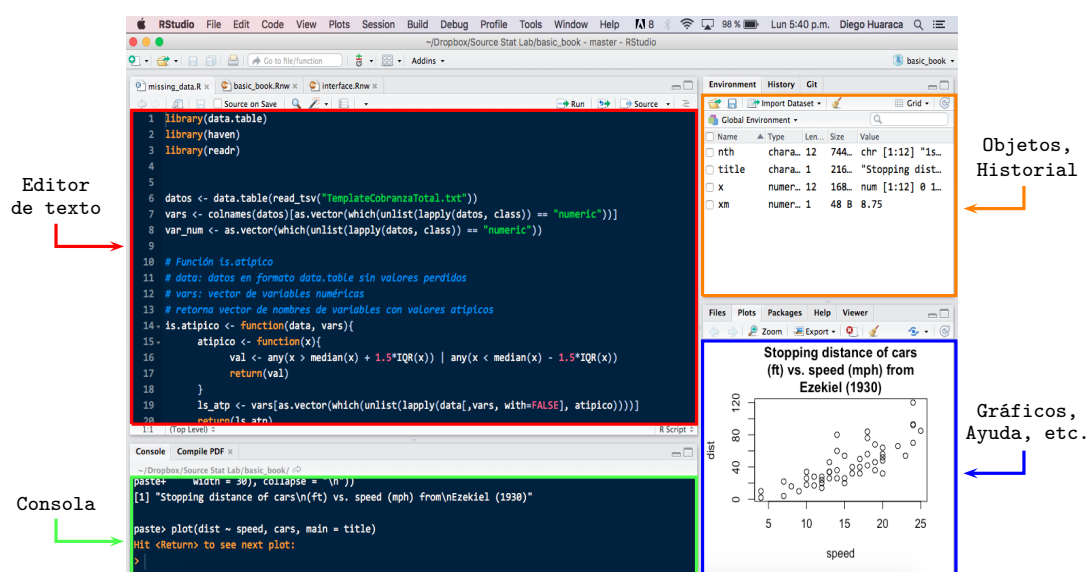


Figura 1.1: Entorno IDE de RStudio

¹Opción válida para la versión server.

RStudio alienta al usuario a hacer uso de las buenas practicas de programación mediante la inclusión de un amplio rango de características haciendo que la experiencia con el programa sea mas eficiente y productiva. Algunas de éstas características son:

- Flexibilidad a la hora de optimizar el espacio de pantalla y rápido retorno visual interactivo.
- Fácil gestión de proyectos e integración con control de versiones.
- Comprobación de código en tiempo real y detección de errores.
- Función de autocompletación de nombres inteligente, utilizada para reducir tiempo en el tecleo de objetos.

Con el propósito que el usuario realice un trabajo rápido y eficiente RStudio incorpora atajos de teclado que reducen la dependencia del ratón, se recomienda a los nuevos usuarios hacer uso de los mismos y adquirir este buen hábito. Para acceder al listado de atajos se debe presionar la combinación de teclas²:

SISTEMA	COMBINACIÓN
Windows / Linux	Alt + Shift + K
Mac OS	Option + Shift + K

1.1.1. Instalación y actualización

RStudio puede ser obtenido libremente desde su página web <http://www.rstudio.org/>. Una vez obtenido el archivo ejecutable la instalación se la realiza de manera simple e intuitiva.

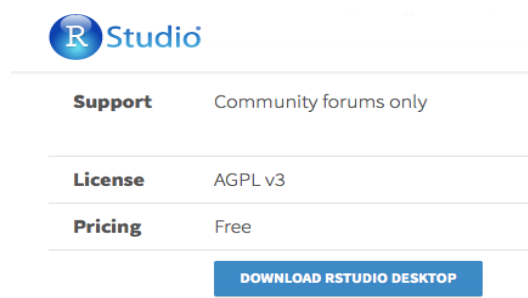


Figura 1.2: Descarga RStudio

Una forma de descarga e instalación más técnica se realiza a partir de los siguientes comandos:

```
# descargar e instalar el paquete installr
install.packages("installr")
# cargar el paquete installr
library(installr)
# instalamos RStudio IDE
install.RStudio()
```

La actualización del programa se realiza desde el menú: Help > Check for Updates.

²En caso de requerir el listado completo de atajos recomendamos visitar la página: <https://support.rstudio.com/hc/en-us/articles/200711853-Keyboards-Shortcuts>.

1.1.2. Funcionamiento

RStudio ofrece una amplia integración con ficheros de diversos formatos: R scripts (.R), Mark-down (.md), LaTeX (.Rnw) entre otros. La facilidad en la generación de documentos dinámicos con RStudio y knitr han hecho que el programa se convierta en la IDE preferida por muchos usuarios de R.

El programa se encuentra organizado en cuatro ventanas de trabajo:

- **Editor de código fuente:** Se encuentra en la zona superior izquierda, esta ventana nos permite abrir y editar ficheros con código R.
- **Consola:** Se ubica en la zona inferior izquierda, esta ventana es también conocida como consola y nos permite ejecutar comandos de R.
- **Navegador de objetos:** La zona superior derecha posee dos ventanas auxiliares:
 - **Workspace:** En esta ventana se enlistan todos los objetos creados en memoria.
 - **History:** En esta ventana se almacena el histórico de las líneas de código que han sido ejecutadas en R.
- **Visualización e información:** Esta última ventana ubicada en la zona inferior derecha se encuentra conformada por 4 ventanas auxiliares:
 - **Files:** Provee el acceso al árbol de directorios y ficheros del disco duro.
 - **Plots:** Ventana auxiliar en la cual aparecen los gráficos creados en la consola.
 - **Packages:** Esta ventana facilita la administración de los paquetes de R instalados en el computador.
 - **Help:** Esta última ventana nos ayuda en la búsqueda de información respecto a un comando en específico.

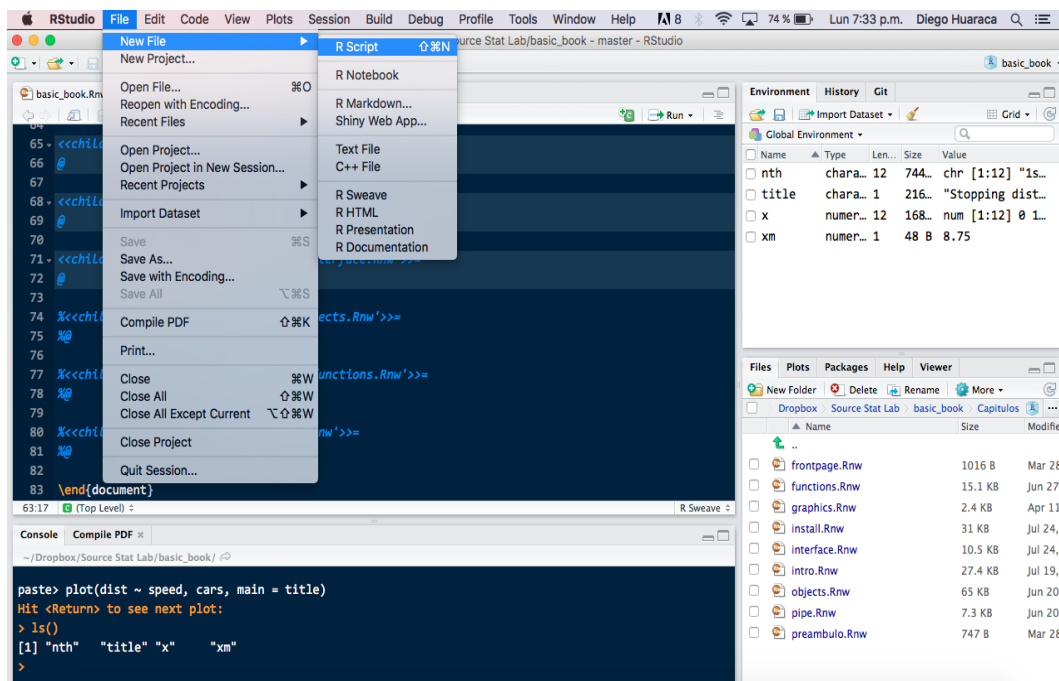


Figura 1.3: Formatos permitidos en RStudio

1.1.3. Proyectos

Para organizar los archivos que empleamos en un trabajo, RStudio nos permite crear proyectos, ésto con el fin de mantener un espacio de trabajo común con todos los archivos asociados o a su vez para compartir en la plataforma Github y tener un control de los cambios realizados sobre los archivos (a este proceso llamaremos *versionamiento*).

Un proyecto puede ser creado en pocos pasos desde el menú principal **File >New Project**:

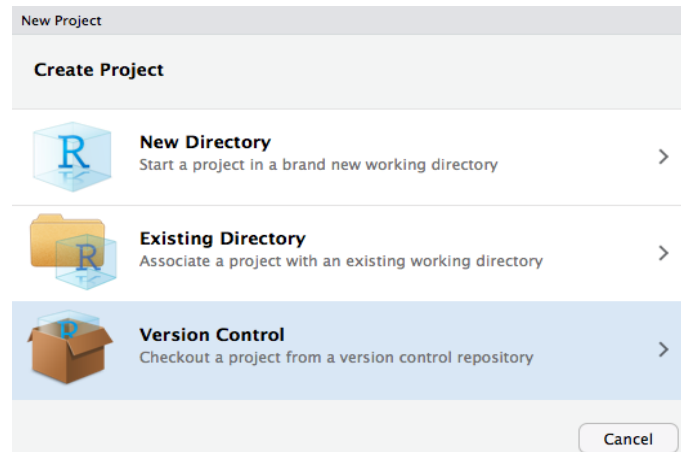


Figura 1.4: Creación de un nuevo proyecto

La primera opción crea un proyecto *sin control de cambios* y con un *nuevo directorio* de trabajo. Para ésta opción se debe asignar un nombre y la ubicación en la que se desea conservar el proyecto.

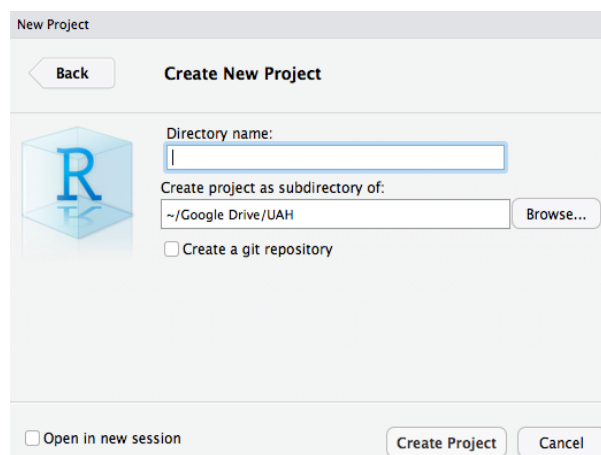


Figura 1.5: Proyecto sin control de cambios 1

La segunda opción crea un proyecto *sin control de cambios* pero fusiona el directorio de trabajo con uno ya existente. Para ésta opción sólo se debe especificar la ubicación del directorio existente.

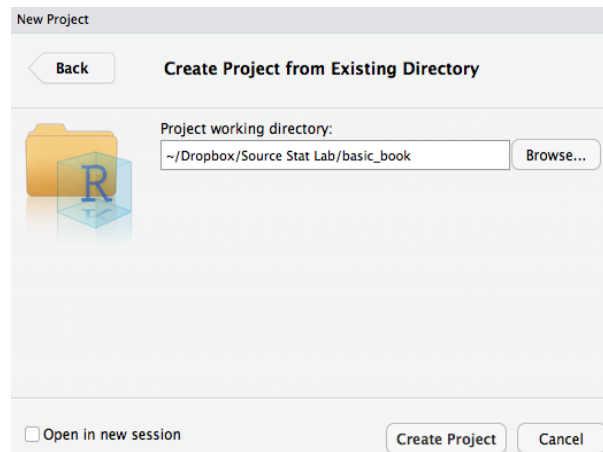


Figura 1.6: Proyecto sin control de cambios 2

La tercera opción permite crear un proyecto *con control de cambios* que pueden ir asociados a los controladores de versiones como: Git o Subversion, se recomienda al usuario hacer uso del primero dada su popularidad y amplia gama de opciones que existe en la actualidad:

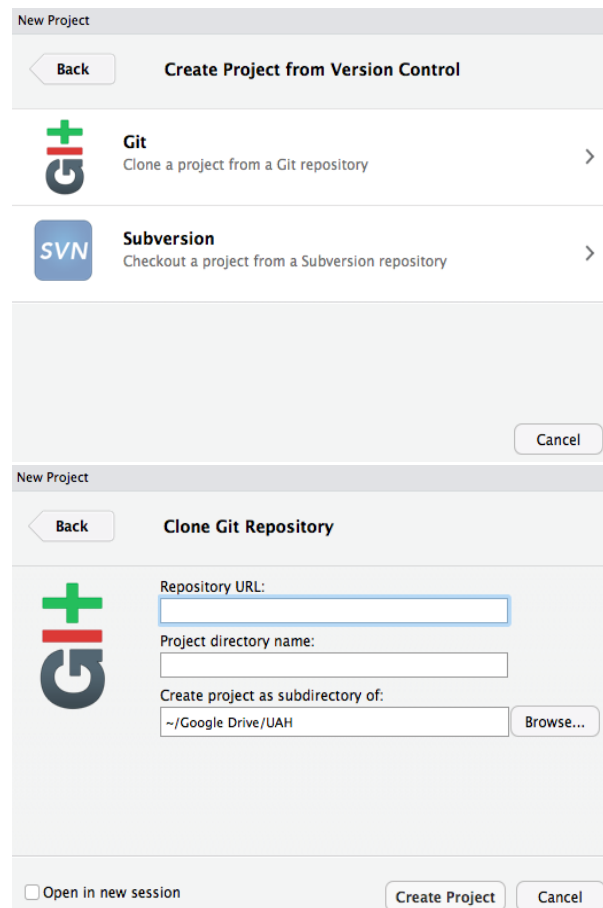


Figura 1.7: Controladores de cambios

Para ésta opción se debe asignar la URL del repositorio de Github con el cual se va a controlar los cambios, un nombre y ubicación para crear el proyecto.

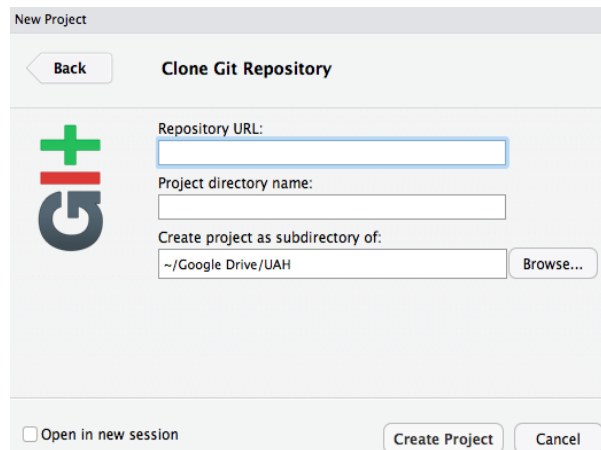


Figura 1.8: Proyecto con versionamiento

La URL del repositorio se obtiene, accediendo a Github

Navegamos fácilmente entre proyectos utilizando un selector (un cuadro combinado) en la barra de herramientas principal ubicada en la esquina superior derecha. El mismo selector tiene una opción para crear un Nuevo Proyecto ..., que elegimos. Para crear un nuevo proyecto, uno completa el nombre y la ubicación del proyecto.

Cuando se crea el proyecto, se establece el directorio de trabajo. La barra de título del panel de la Consola se actualiza, al igual que el contenido del componente Archivos, que enumera los archivos y subdirectorios en un directorio determinado. El componente Archivos reside en un bloc de notas, que de forma predeterminada se coloca en la esquina superior derecha. Si no se muestra, selecciona su pestaña. En la Figura 2-1, vemos que nuestro directorio de trabajo contiene nuestro archivo de datos y un archivo de contabilidad que RStudio creó.

RStudio ofrece varios mecanismos para controlar varios aspectos de la evaluación durante una sesión. La función `options()` es empleada para compartir los valores de parámetros entre las funciones.

1.1.4. Ancho de impresión

El usuario puede controlar el ancho de impresión de los resultados que se muestran en la pantalla, modificando el parámetro `width`. El siguiente comando muestra el ancho de impresión actual:

```
getOption("width")
```

```
## [1] 75
```

Una vez conocido el ancho de pantalla se procede a modificar el mismo cambiando el valor del parámetro `width`, de la siguiente manera:

```
options(width=30)
```

```
rnorm(8)
```

```
## [1] 0.29039754 0.24355520
## [3] 0.06626656 -0.85325739
## [5] -0.09195775 1.39533481
## [7] 1.51375331 -0.61084817
```

```
options(width=55)
rnorm(8)

## [1]  0.821149525  0.499520211  0.005248789 -0.817091531
## [5]  0.038566898 -0.376564684  1.798921238  0.213625777
```

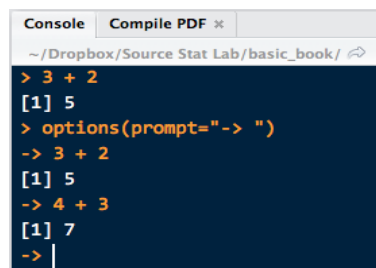
```
options(width=70)
rnorm(8)

## [1] -0.9631736 -1.1384812 -0.8862787  0.5466695  0.0947083  0.5745882
## [7]  1.1152412 -1.1603917
```

1.1.5. Prompt

Si el usuario desea cambiar el símbolo `>` del prompt o interpretador por otro símbolo diferente, por ejemplo: `→` o por un nombre, debe modificar el parámetro `prompt`:

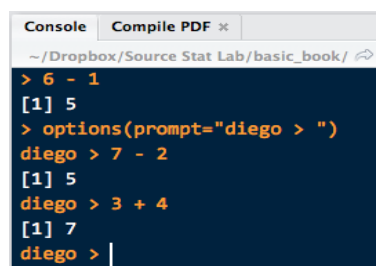
```
options(prompt="-> ")
```



```
Console  Compile PDF ✕
~/Dropbox/Source Stat Lab/basic_book/ ↗
> 3 + 2
[1] 5
> options(prompt="-> ")
-> 3 + 2
[1] 5
-> 4 + 3
[1] 7
-> |
```

Figura 1.9: Modificación del prompt a `→`

```
options(prompt="diego >")
```



```
Console  Compile PDF ✕
~/Dropbox/Source Stat Lab/basic_book/ ↗
> 6 - 1
[1] 5
> options(prompt="diego > ")
diego > 7 - 2
[1] 5
diego > 3 + 4
[1] 7
diego > |
```

Figura 1.10: Modificación del prompt a `diego >`

1.1.6. Decimales

Una preocupación adicional para los usuarios es la cantidad de decimales con la cual se muestran los resultados, dicha cantidad de decimales puede ser modificada y debe encontrarse en el rango de 1 a 22.

```
getOption("digits")
```

```
## [1] 7
```

R por default muestran los resultados con 7 decimales, sin embargo los mismos pueden ser modificados como se muestra a continuación:

```
options(digits=2)
```

```
rnorm(3)
```

```
## [1] -0.80 -1.10 -0.12
```

```
options(digits=5)
```

```
rnorm(3)
```

```
## [1] -2.425510 -0.031174 -1.688828
```

```
options(digits=10)
```

```
rnorm(3)
```

```
## [1] 1.4404550176 0.8232495876 -0.1863492083
```

Existen opciones adicionales que pueden ser modificadas de acuerdo a las necesidades que tenga el usuario, para ver el listado completo de opciones podemos teclear en la consola el comando:

```
help(options)
```

1.1.7. Respaldo de información

Un tema importante dentro del análisis de datos es el respaldo de información que se pueda dar sobre ciertos resultados obtenidos, en este punto R consta de dos comandos muy útiles: `save()` & `load()`.

El primero de ellos permite almacenar en disco los objetos que desee el usuario (almacenamiento parcial), dicho comando también puede ser configurado de tal manera que almacene todos los objetos que se encuentra válidos en el área de trabajo.

```
# si deseamos guardar el objeto "datos_banco" con el nombre "base"
```

```
save(datos_banco, file = "base.RData")
```

```
# para el caso que se desee almacenar todos los objetos con el nombre "info"
```

```
save(list = ls(all = TRUE), file = "info.RData")
```

El segundo comando nos va a permitir cargar los objetos guardados en el área de trabajo actual o en un ambiente determinado.

```
# cargamos el objeto base en el area de trabajo
```

```
load("base.RData")
```

```
# ahora cargamos "info" en un ambiente determinado "env"
```

```
load("info.RData", envir = env)
```

1.2. R Analytic Flow

R Analytic Flow (RAF) es una interfaz gráfica de usuario desarrollado por Ryota Suzuki³, que facilita el análisis de datos a través de diagramas de flujo. El software se encuentra bajo licencia BSD & GPL, por lo cual puede obtenerse de forma gratuita a través de la página web de Ef-prime, Inc. <http://www.ef-prime.com> para las plataformas: Windows, Mac OS X y Linux.

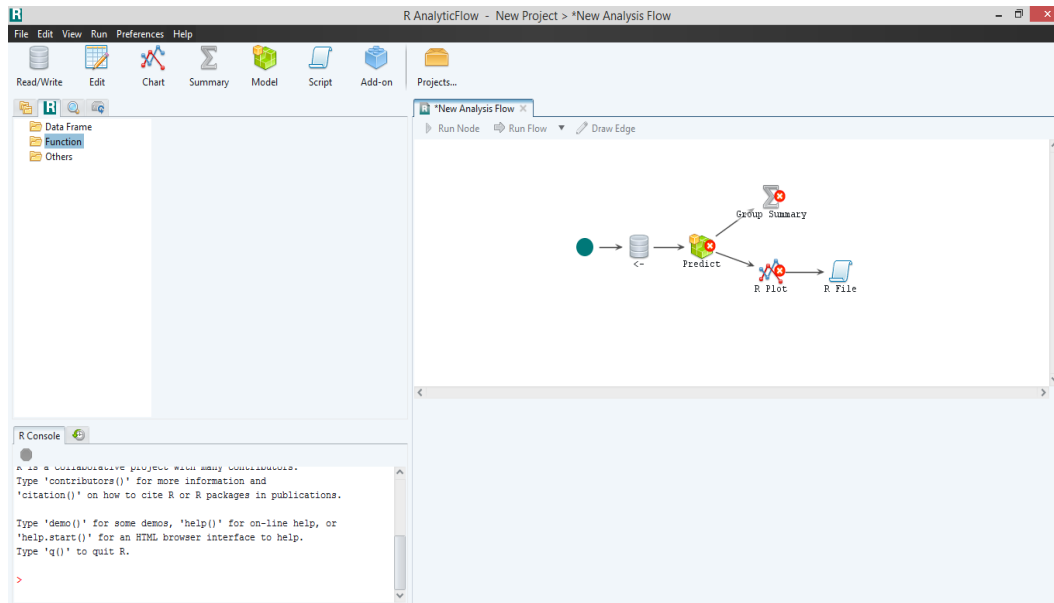


Figura 1.11: R Analytic Flow

1.2.1. Ventajas

A continuación enumeramos algunas de las ventajas que posee R Analytic Flow:

1. Facilita ejecutar procesos a través de flujos.
2. Fácil implementación de tareas en cada nodo.
3. Reduce la complejidad a la hora de programar varias funciones que se relacionen entre sí.
4. El número de usuarios que usan R Analytic Flow va en aumento debido a las facilidades que presenta.

1.2.2. Desventajas

Algunas de las desventajas por las cuales los usuarios no usan R Analytic Flow:

1. Escasa documentación sobre el manejo de la interfaz.
2. El código fuente se encuentra administrado únicamente por Ef-prime, Inc. Esto impide que se pueda seguir optimizando la interfaz con mayor rapidez.

³Ryota Suzuki es un desarrollador de software orientado al análisis de datos, fundó con sus amigos la empresa Ef-prime, Inc. en Tokyo, además es el creador del paquete *pvcust* de R.

1.2.3. Funcionamiento

R Analytic Flow al ser un software de análisis de datos que emplea el entorno R como su motor organiza sus procesos de análisis mediante flujos de trabajo. Todos los procesos pueden ser reproducidos de forma sencilla y precisa simplemente haciendo uso del ratón.

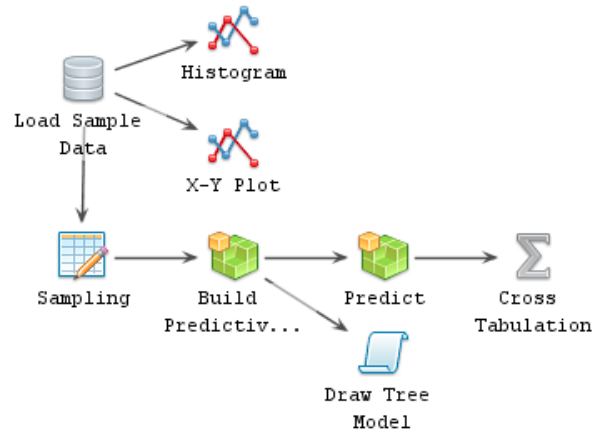


Figura 1.12: Flujo de trabajo

En general, los flujos de trabajo se combinan con datos y documentos relacionados para formar en sí un proyecto.