

Joint Jack Mackarel Model: Guía de uso

Wencheng Lau-Medrano

Ricardo Oliveros-Ramos

Instituto del Mar del Perú

versión 0.1, 7 de septiembre de 2014

Índice

1. Prerrequisitos	2
1.1. Instalación de ADMB	2
1.2. Instalación de EMACS	8
1.3. Instalación de R y RStudio	9
1.3.1. Instalación de R	9
1.3.2. Instalación de RStudio	11
2. Instalación de JJM	15
2.1. ¿Cómo obtener una copia desde GitHub?	15
2.1.1. Descargar una imagen de JJM	15
2.1.2. Sincronizar una carpeta con el repositorio	17
2.2. Compilación del modelo	21
2.2.1. Compilación desde EMACS	21
3. Ejecución de JJM	23
3.1. Archivos necesarios	23
3.2. Ejecución	25
4. Visualización de resultados	27
4.1. Descripción de los scripts de R	27
4.2. Ejecución de los scripts	28
4.2.1. Pasos previos	28
4.2.2. Segundo script	30

1. Prerrequisitos

En esta sección se darán las pautas previas a la ejecución del modelo. Para dicho fin, se hará uso de dos lenguajes de programación (**C++** y **R**) por lo que, del mismo modo, se utilizarán dos softwares relacionados a cada uno de ellos (**EMACS** y **RStudio**).

1.1. Instalación de ADMB

Emacs es un editor de texto con una gran cantidad de funciones, muy popular entre programadores y usuarios técnicos. Dentro de nuestros intereses, es el programa que usaremos para abrir, editar y compilar el código de **ADMB** (hecho en **C++**).

Para la instalación debemos:

1. Descargar el programa **Rapid Environment Editor** desde <http://www.rapidee.com/en/download> e instalarlo.
2. Ir al website <http://www.admb-project.org/tools/admb-tools-for-windows> y seleccionar **ADMB Tools for nn-bit Windows**, donde **nn** (32 o 64) depende del ordenador en donde se está trabajando.
3. Este link, llevará a descargar el archivo **ADMBtoolsnn-vv.7z**, donde **vv** es el número de versión.
4. Retornar al website <http://www.admb-project.org/tools/admb-tools-for-windows> y seleccionar el link **ADMB vv.v.rrrr binary library for nn-bit Windows**, donde **vv.v** es el número de versión y **rrrr** es el número de revisión de ADMB.
5. Ir al directorio donde se descargaron ambos archivos **.7z**. Dar click derecho en el archivo **ADMBtools** y seleccionar **Extraer aquí**.

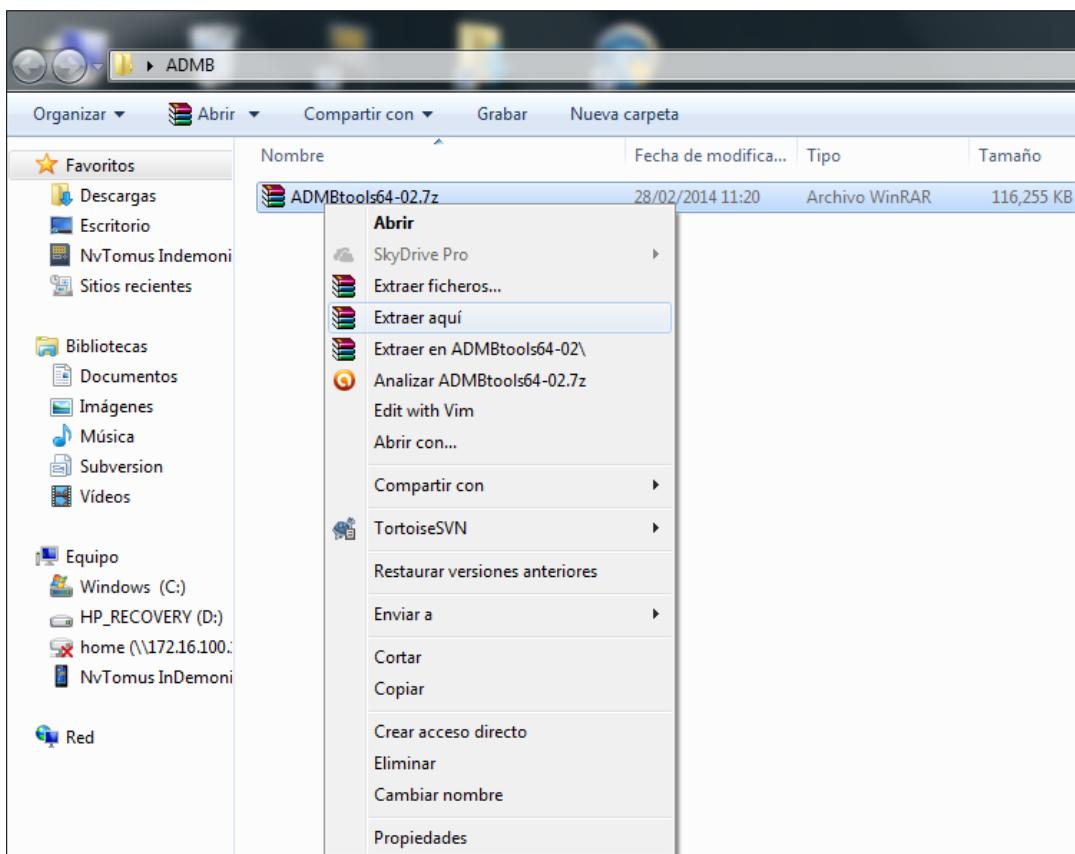


Figura 1: Descompresión del archivo **ADMBtools64-02.7z**

6. Una vez completada la extracción, la carpeta obtenida contendrá una carpeta llamada **mingw**. Copiar (o cortar-pegar) **mingw** a la locación deseada. Para este ejemplo, la locación será C:\. Por lo que, finalmente, se tendrá C:\mingw.
7. Ir al directorio donde se descargaron ambos archivos **.7z**. Dar click derecho en el archivo **ADMBlib64.11.1.1483.7z** y seleccionar **Extraer aquí**.

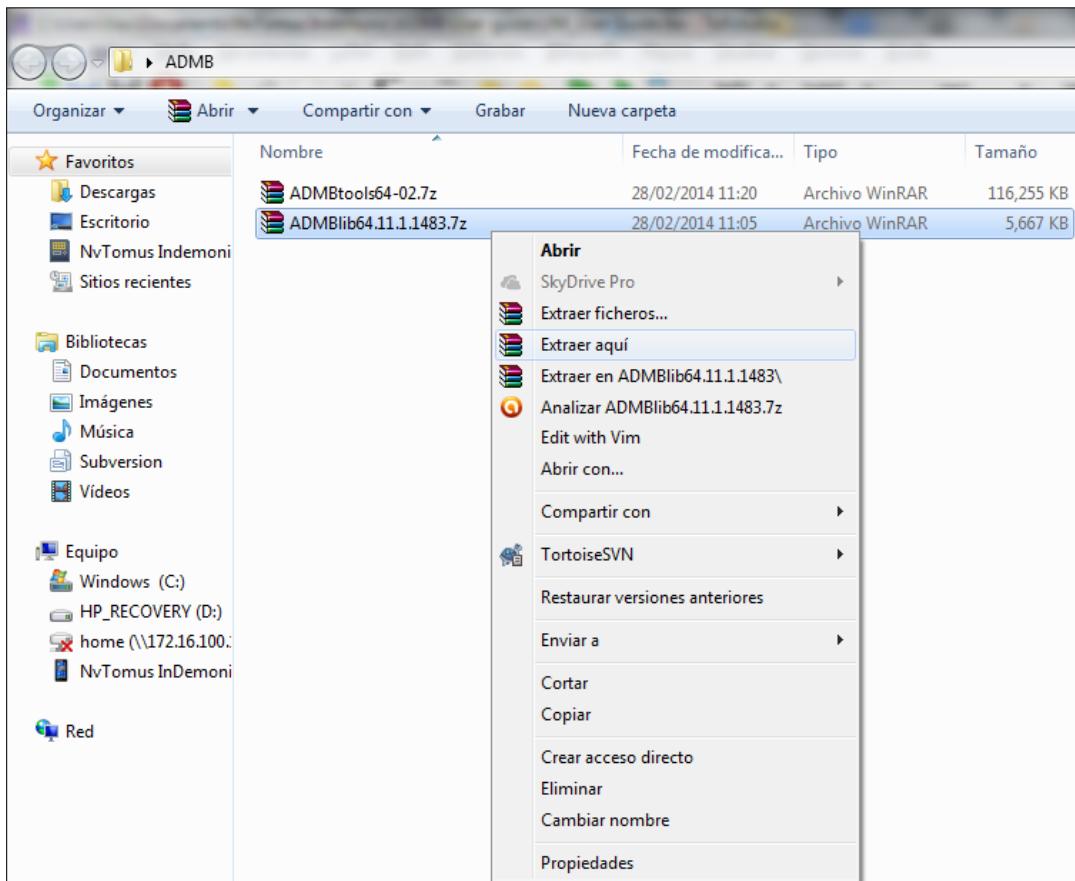


Figura 2: Descompresión del archivo **ADMBlib64.11.1.1483.7z**

8. Una vez completada la extracción, renombrar la carpeta con el nombre **admb** y copiar (o cortar-pegar) a la locación deseada. Para este ejemplo, la locación será C:\. Por lo que, finalmente, se tendrá C:\admb.

9. Seguidamente, es momento de decirle a Windows dónde se encuentran los archivos. Para esto se utilizará el programa **Rapid Environment Editor** (citado anteriormente). Ir a *Inicio → Programas → RapidEnvironmentEditor → RapidEnvironmentEditor*. Ejecutar como administrador dando click derecho en el Acceso directo.

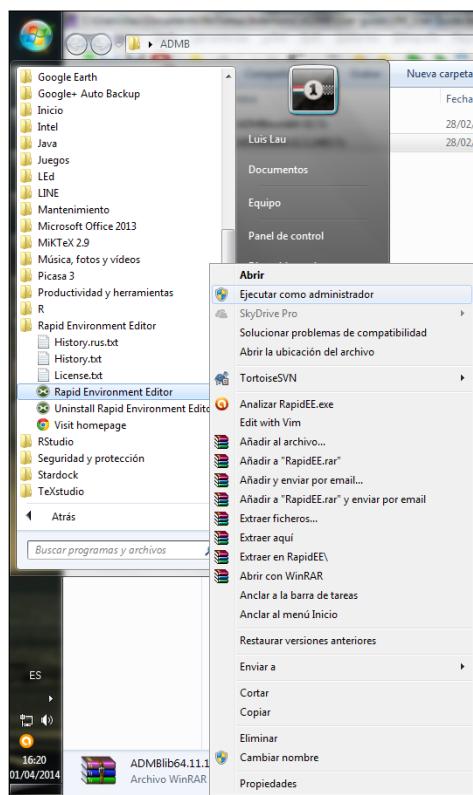


Figura 3: Abrir **Rapid Environment Editor** en modo Administrador

10. Buscar la línea **Path**, dar click derecho y seleccionar la opción **Añadir valor**

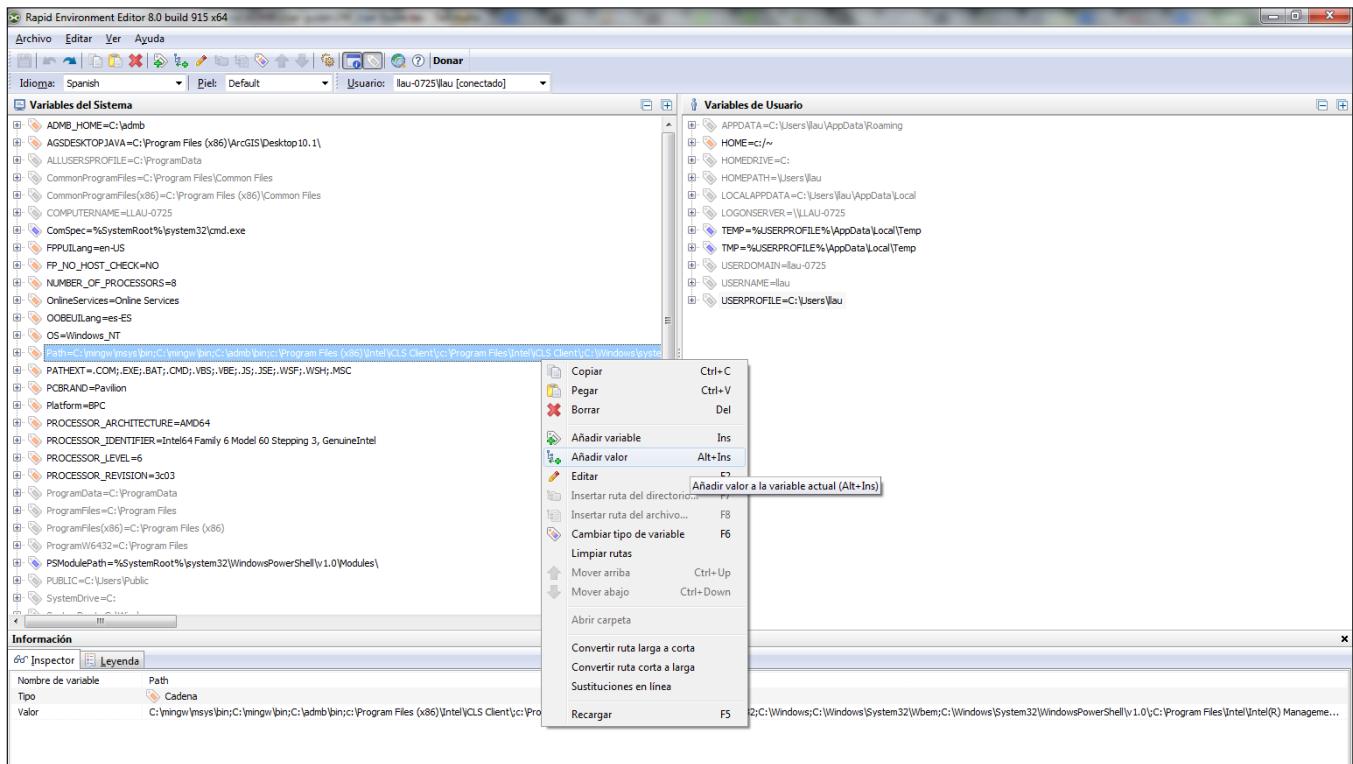


Figura 4: Añadir valor

11. Seleccionar la opción **Insertar ruta del directorio**

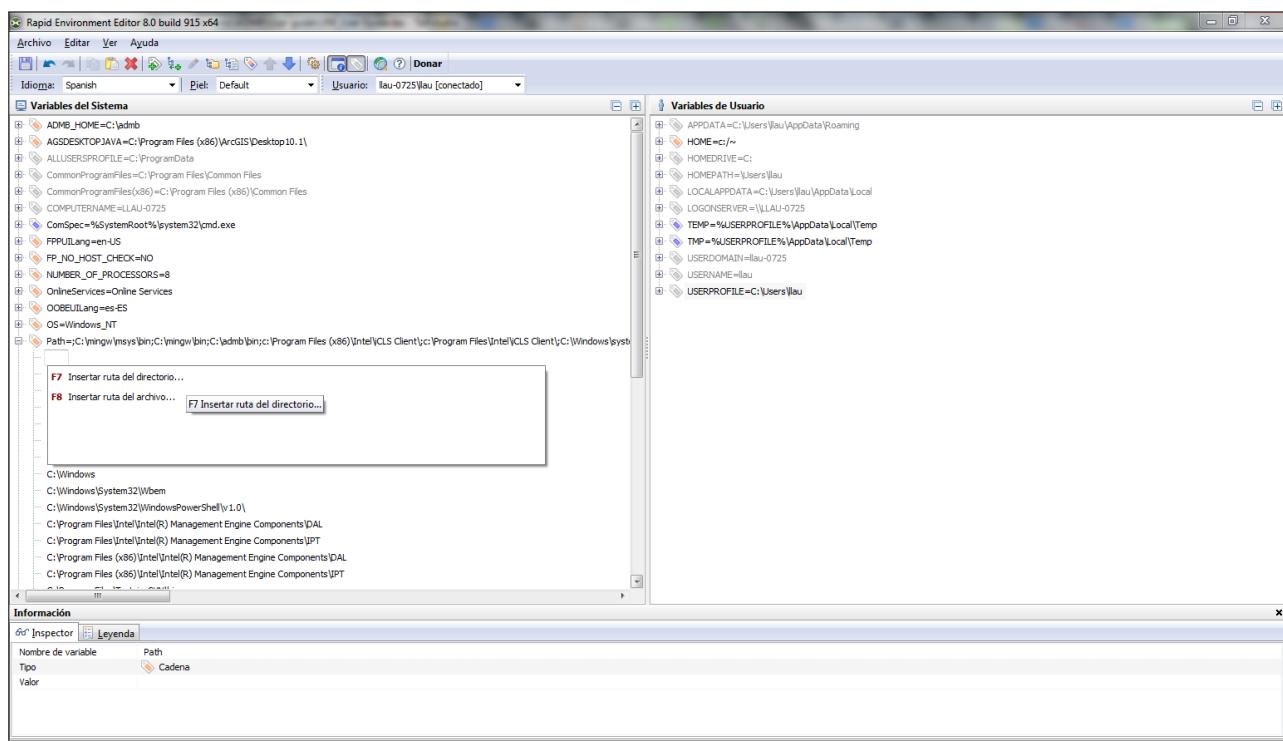


Figura 5: Insertar ruta del directorio

12. Mediante la ventana abierta, seleccionar la carpeta C:\admb\bin

13. Realizar el mismo procedimiento para las siguientes carpetas: C:\mingw\bin, C:\mingw\msys\bin

14. Luego, ir a **Editar**, opción **Aadir variable**

15. En la ventana que se abre, escribir **ADMB_HOME** y **Aceptar**.

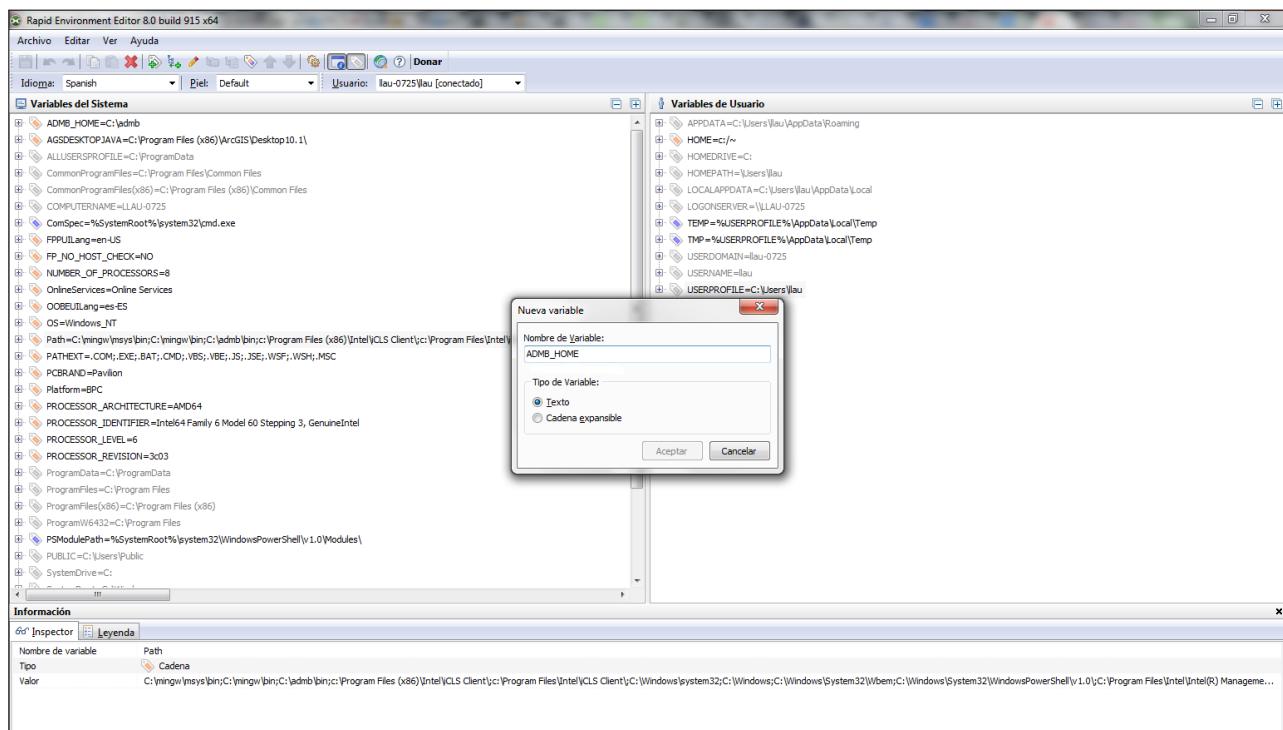


Figura 6: Aadir variable **ADMB_HOME**

16. Seguidamente, ir al menú **Archivo** y seleccionar la opción **Guardar**. Cerrar **Rapid Environment Editor**.

17. Para probar que la instalación haya sido bien realizada, ir hasta la carpeta C:\admb\examples\admb\simple
18. Copiar los archivos simple.tpl y simple.dat a una carpeta externa (dentro Escritorio, por ejemplo).
19. Abrir una ventana de comandos (**DOS**) en la carpeta elegida. En nuestro ejemplo, crearemos una carpeta en Escritorio llamada **ADMB** y es aquí donde copiaremos los comandos del paso anterior. Para abrir la ventana de comandos, presionar **Shift + Click derecho** sobre la carpeta y elegir la opción **Abrir ventana de comandos aquí**.

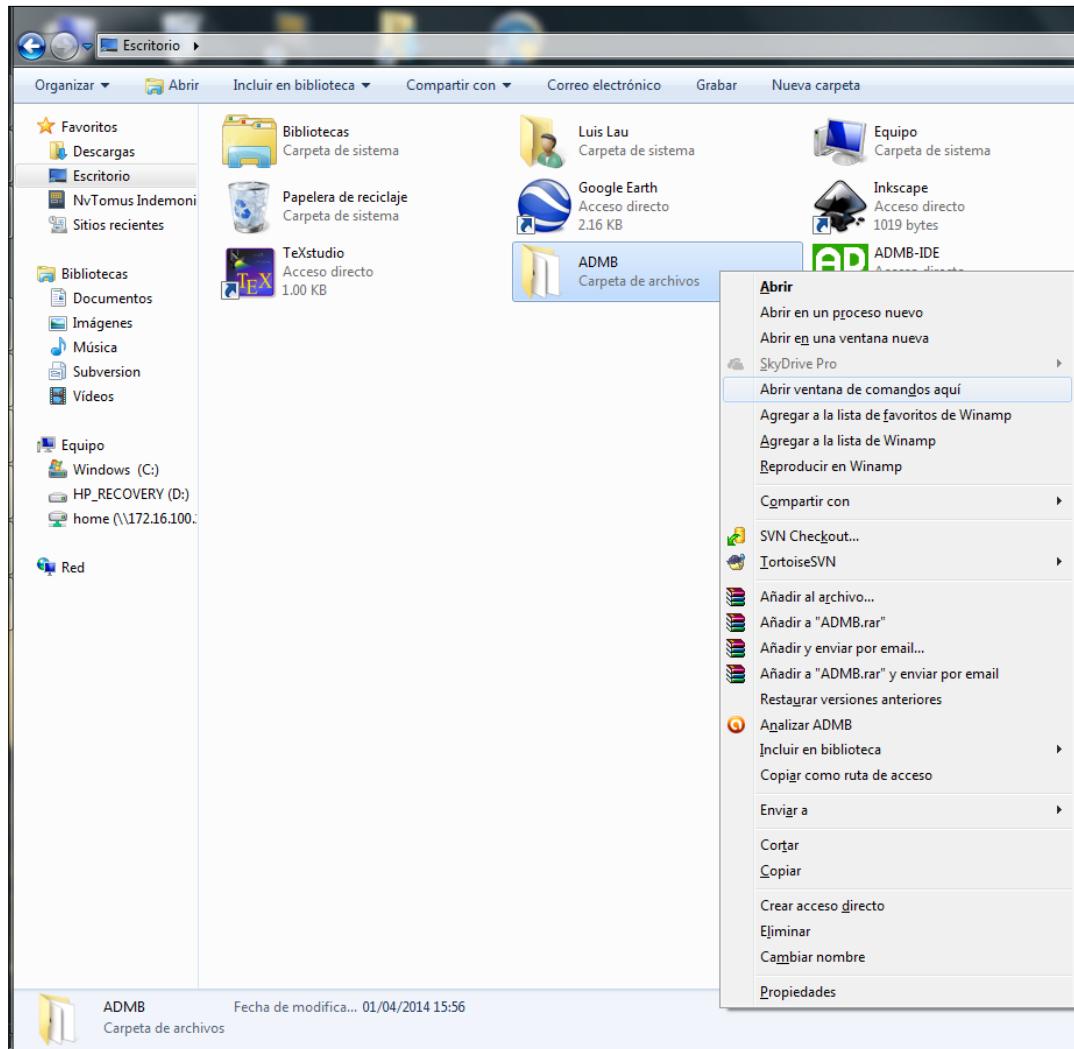


Figura 7: Abrir ventana de comandos en carpeta seleccionada

20. En la ventana (símbolo de sistema) que se abrirá, escribir el siguiente comando: `admb simple.tpl` y dar **Enter**.

21. Si la instalación fue exitosa, el resultado debería ser:

```
*** Parsing simple.tpl:  
*** Compile simple.cpp:  
*** Linking simple.obj:  
Successfully built 'simple.exe'
```

22. Luego, en la misma ventana de Símbolo de sistema, escribir `simple simple.tpl` y dar **Enter**.

23. Si la instalación fue exitosa, el resultado debería ser:

```
Initial statistics: 2 variables; iteration 0; function evaluation 0; phase 1  
Function value 3.6493579e+001; maximum gradient component mag -3.6127e+000  
Var Value Gradient | Var Value Gradient | Var Value Gradient  
1 0.00000 -3.61269e+000 | 2 0.00000 -7.27814e-001 |  
  
- final statistics: 2 variables; iteration 7; function evaluation 19  
Function value 1.4964e+001; maximum gradient component mag -7.0014e-005  
Exit code = 1; converg criter 1.0000e-004  
Var Value Gradient | Var Value Gradient | Var Value Gradient  
1 1.90909 -7.00140e-005 | 2 4.07818 -2.08982e-005 |  
  
Estimating row 1 out of 2 for hessian  
Estimating row 2 out of 2 for hessian
```

1.2. Instalación de EMACS

- Ir al website: <http://www.admb-project.org/tools/admb-ide> y dar click en Download



Figura 8: Web de ADMB

- Ejecutar el instalador con las opciones por defecto.

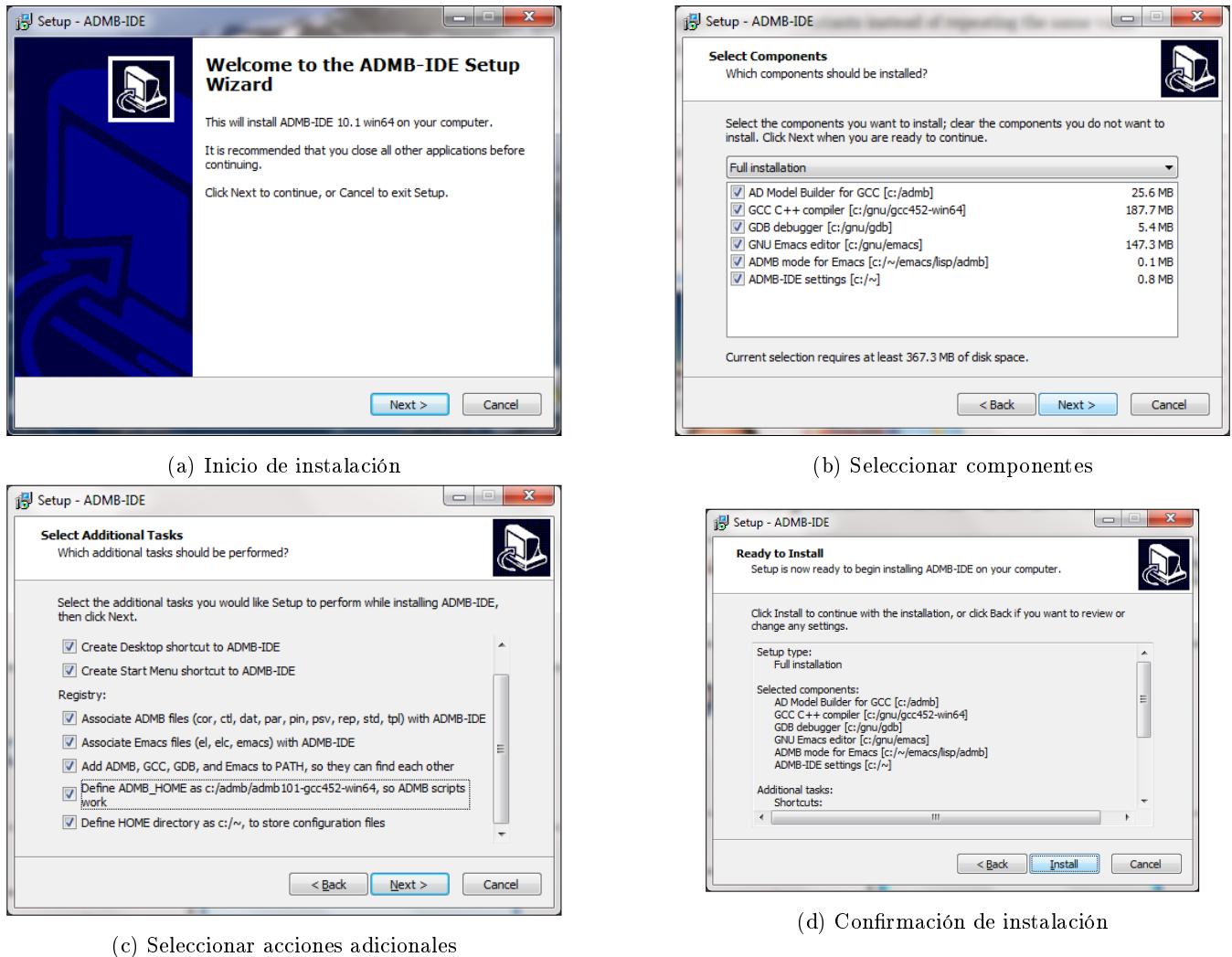


Figura 9: Proceso de instalación ADMB.

1.3. Instalación de R y RStudio

1.3.1. Instalación de R

Para obtener los resultados en JJM, se debe utilizar R 3.0.0 o superior. Instalar R de la siguiente manera.

1. Descargar la última versión de R desde AQUÍ.

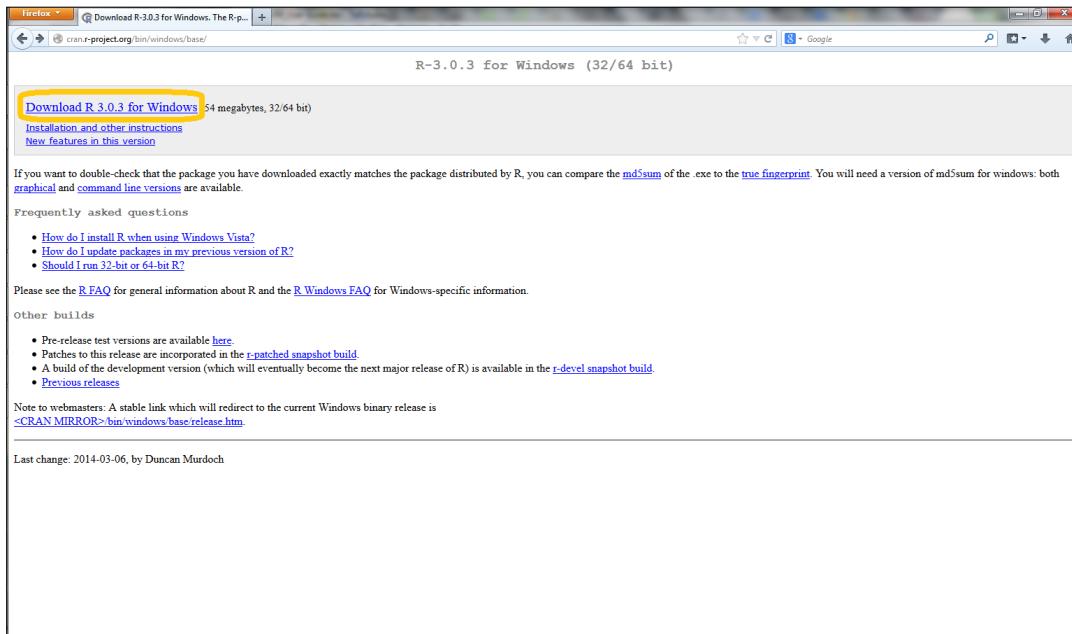


Figura 10: Web de RStudio

2. Hacer doble clic sobre el fichero para iniciar la instalación de R.
3. Elegir el idioma de instalación y a continuación arrancará el asistente de instalación, dar clic en **Siguiente**.
4. Aparecerá una ventana en la cual se solicita al usuario leer información relacionada a la licencia. Al finalizar esta lectura, dar clic en **Siguiente**.
5. A continuación aparecerá una ventana en la que se indica la ruta de instalación, dar clic en **Siguiente**.
6. La instalación permitirá que el usuario seleccione qué componentes desea instalar.
7. Seguidamente, en la siguiente ventana **Opciones de configuración**, el usuario podrá hacer uso de opciones de avanzadas de instalación. En caso de querer usar una instalación típica dar clic en **No** y luego en **Siguiente**.
8. Luego se debe elegir la carpeta del menú inicio donde se colocará los accesos directos del programa, dar clic en **Siguiente**.
9. Seleccionar las tareas adicionales y posteriormente dar clic en **Siguiente**.
10. A continuación se inicia la instalación de R. Una vez finalizado el proceso, aparecerá una ventana que indicará la culminación del mismo.

La secuencia de pasos para el proceso de instalación de R son mostrados en la Figura 11.

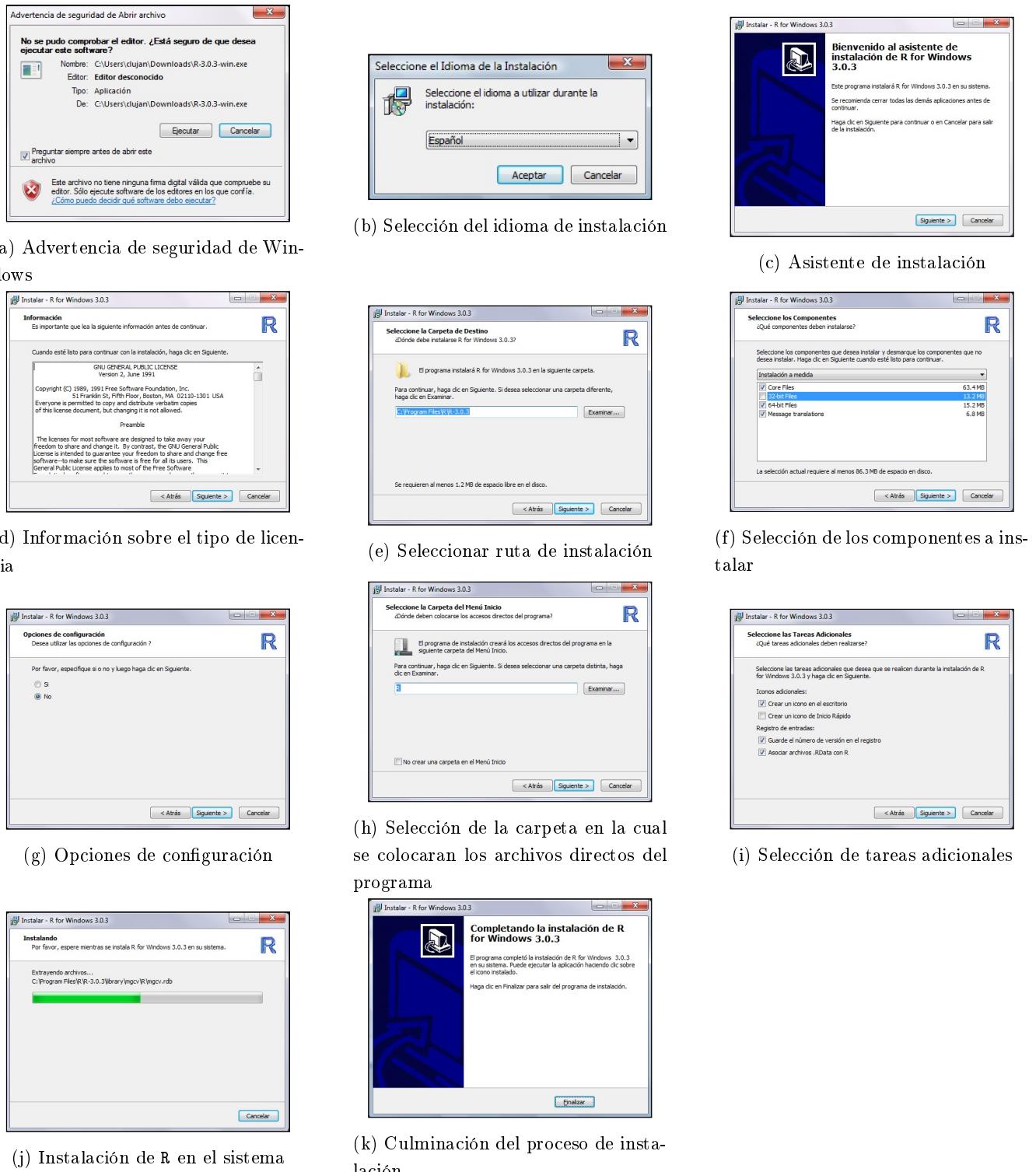


Figura 11: Proceso de instalación de R.

1.3.2. Instalación de RStudio

RStudio es una interface de usuario para R. Es libre y de código abierto (Open Source) y se encuentra disponible para los sistemas operativos Windows, Mac y Linux.

Se detallará a continuación los pasos para la descarga e instalación de RStudio para entorno Windows.

1. Descargar la última versión de RStudio desde AQUÍ.

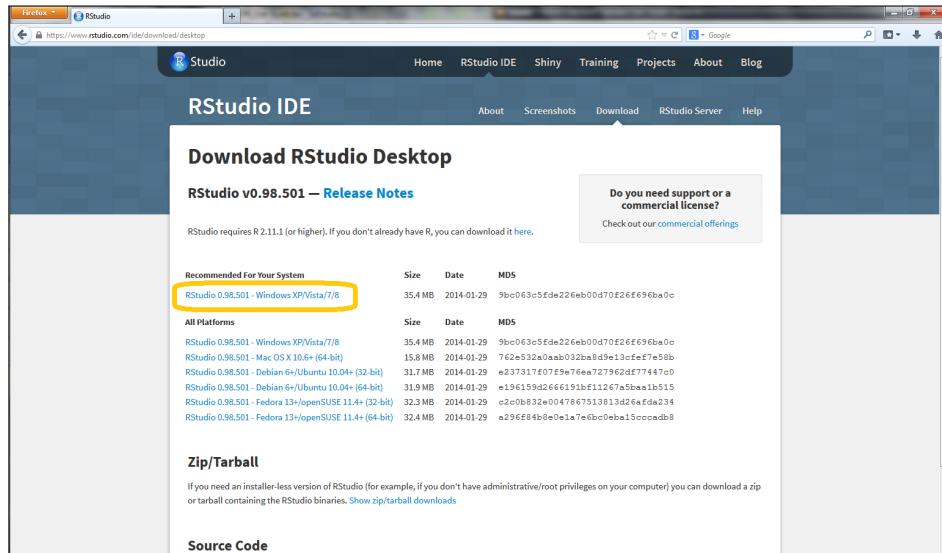


Figura 12: Web de RStudio

2. Descargar.

(a) Sitio oficial de RStudio

(b) Descargar instalador para desktop

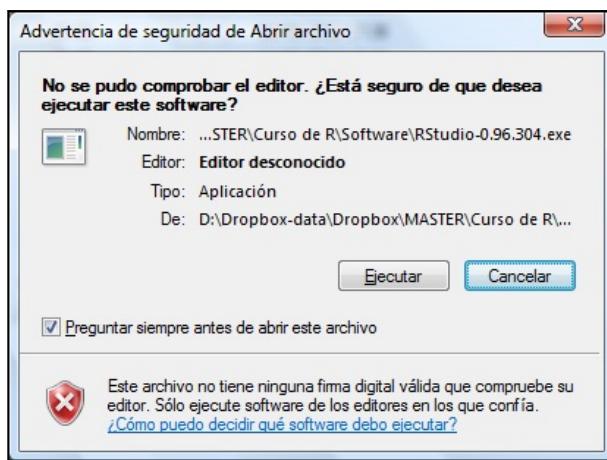
(c) Seleccionar el instalador en relación a sistema operativo

(d) Seleccionar la carpeta en la cual se guardará el instalador

Figura 13: Iniciando el proceso de descarga del instalador del RStudio.

3. Ejecutar el instalador descargado. De manera similar a la instalación de R, saldrá un aviso de Windows indicando que no se puede comprobar el editor, ignorar éste aviso y dar clic en **Ejecutar**. Puede aparecer otra ventana indicando que hay que permitir la ejecución del software, dar clic en **Permitir**
4. A continuación aparecerá el asistente de instalación, dar clic en **Siguiente**.
5. Indicar el lugar de instalación del software, dar clic en **Siguiente**.
6. Elegir la carpeta del menú inicio donde se colocarán los accesos directos a los elementos del software, finalizando este paso comienza la instalación del software.
7. Este último paso dura pocos segundos y al finalizar la instalación aparecerá el asistente de instalación, indicando que el proceso ha concluido, dar clic en **Terminar**.

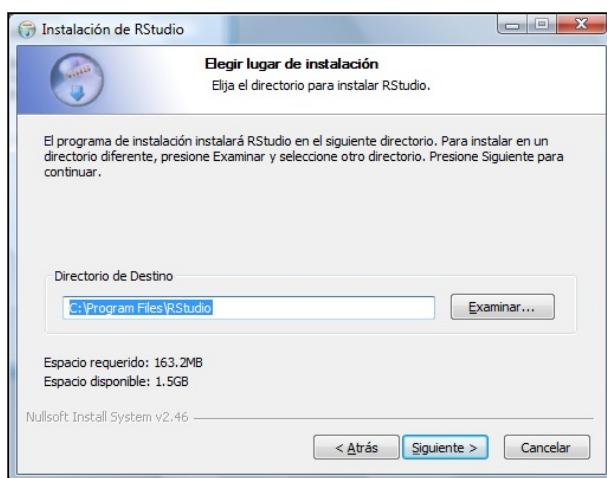
La secuencia de pasos para el proceso de instalación de RStudio son mostrados en la Figura 14.



(a) Advertencia de seguridad de Windows



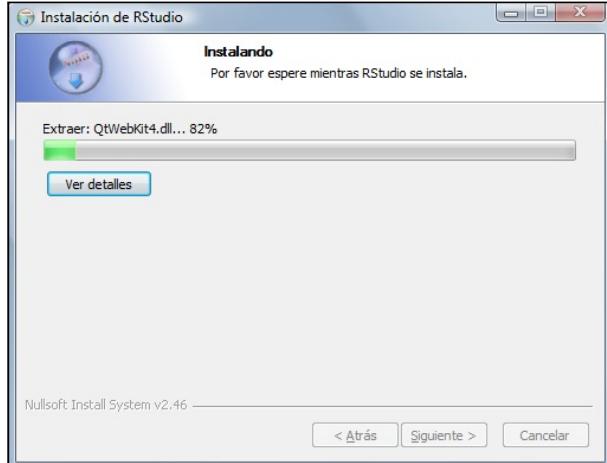
(b) Asistente de instalación



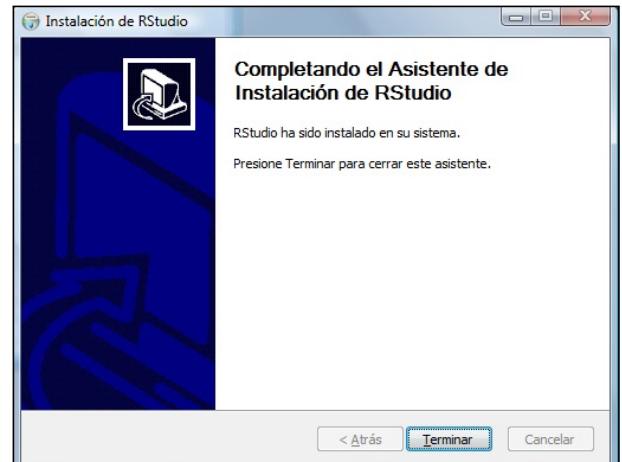
(c) Seleccionar ruta de instalación



(d) Selección de la carpeta en la cual se colocaran los archivos directos del programa



(e) Instalación de RStudio en el sistema



(f) Culminación del proceso de instalación

Figura 14: Proceso de instalación de RStudio.

Al acceder a RStudio, se observará una ventana subdividida tal cual se muestra en la Figura 15. En la parte izquierda se hallará la **consola**; en la superior derecha, el espacio de trabajo (**Workspace**) y el historial de trabajo (**History**); en la inferior derecha, cuatro pestas?: **Files**, **Plots**, **Packages**, **Help** y **Viewer**. Finalmente, en la parte superior de la ventana se encuentra la barra de herramientas.

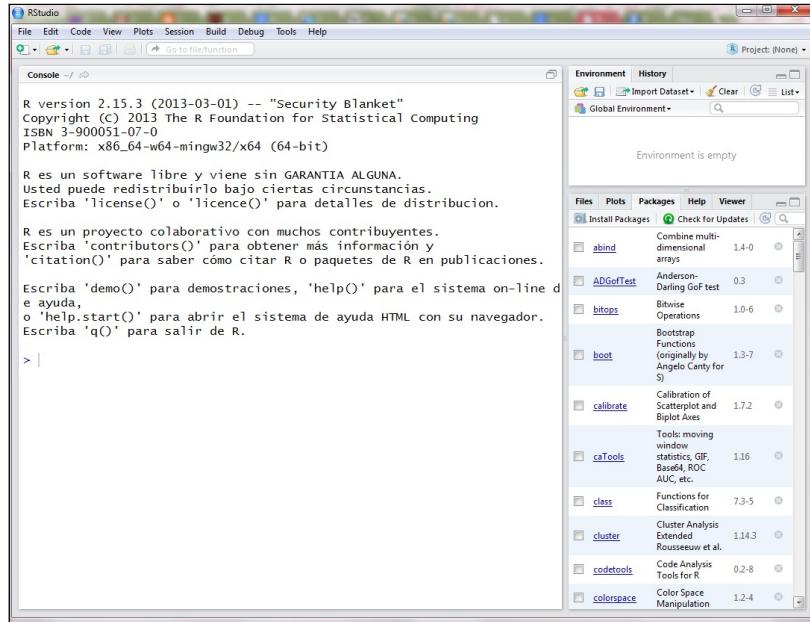


Figura 15: Ventana principal de RStudio.

2. Instalación de JJM

2.1. ¿Cómo obtener una copia desde GitHub?

El código necesario para ejecutar JJM se encuentra alojado en el servidor GitHub. Para descargar una imagen almacenada o para sincronizar se hará uso del programa TortoiseGit.

Descargar TortoiseGit desde <https://code.google.com/p/tortoisegit/wiki/Download?tm=2> e instalar.

2.1.1. Descargar una imagen de JJM

1. Acceder al repositorio desde el siguiente link: https://github.com/SPRFMO/jack_mackerel.
2. Ir al botón **Copy to clipboard** (localizado en la parte inferior de la columna derecha).

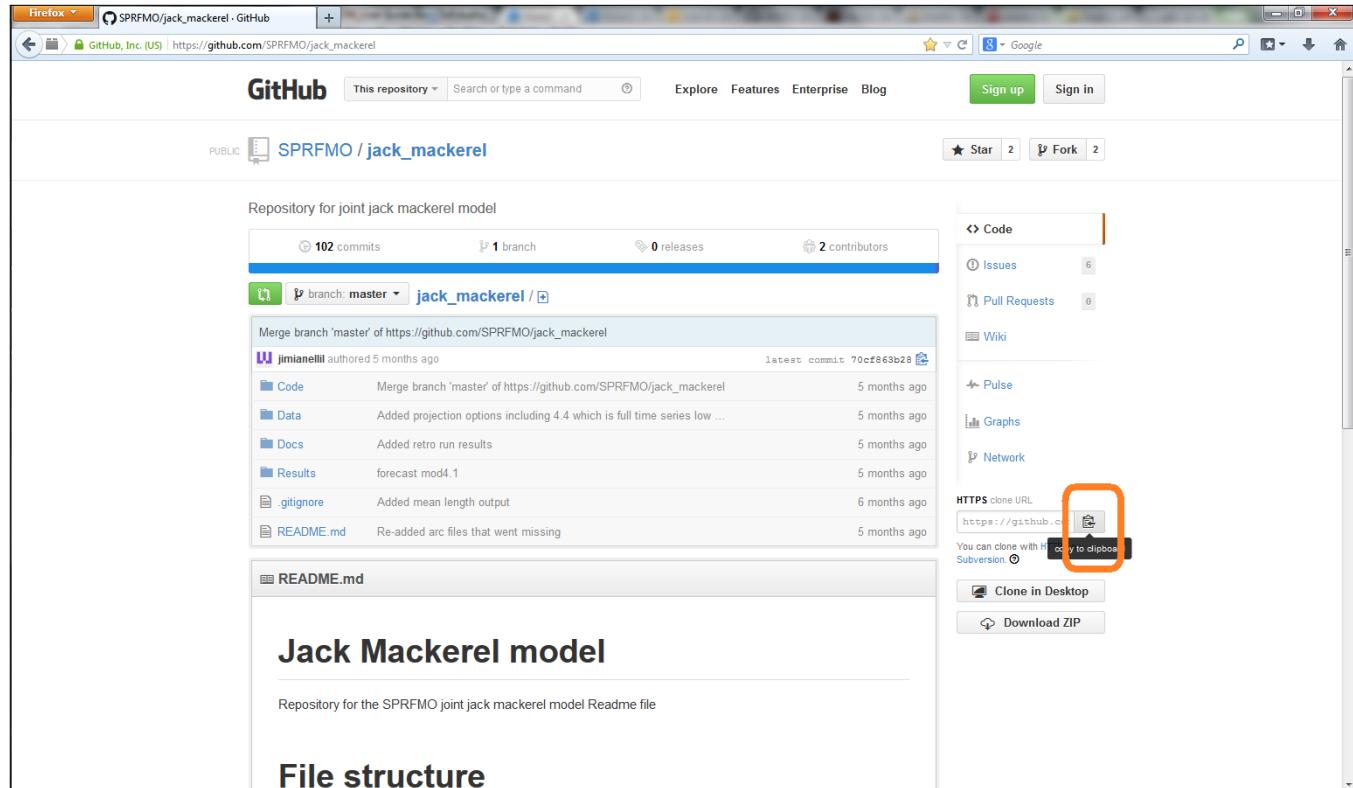


Figura 16: Web del repositorio de ADMB en GitHub

3. Crear carpeta donde se desea realizar la copia con el nombre SPRFM0_SC. Por ejemplo: Escritorio\SPRFM0_SC
4. Dar click derecho dentro de la carpeta y seleccionar la opción **Git Clone**

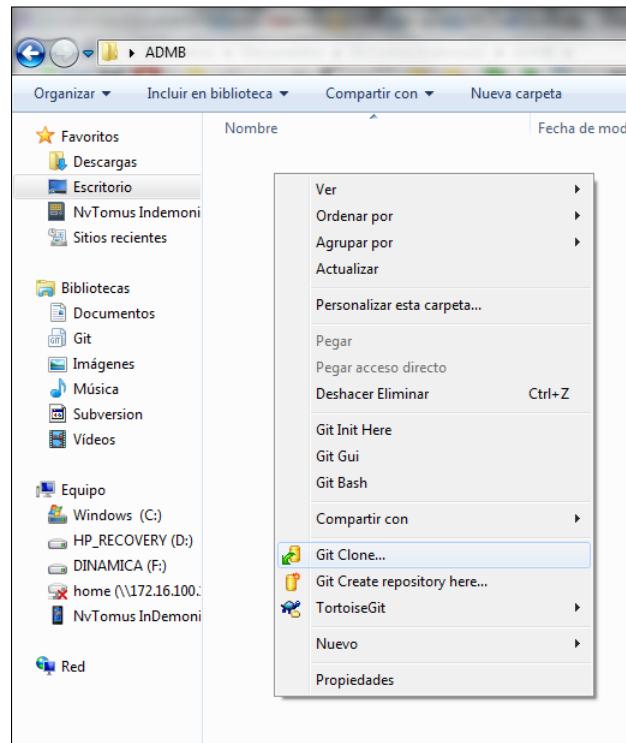
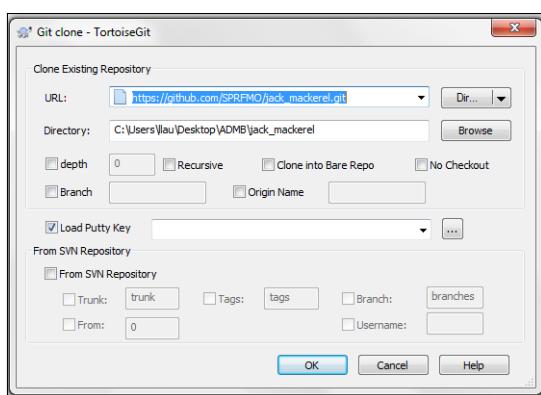
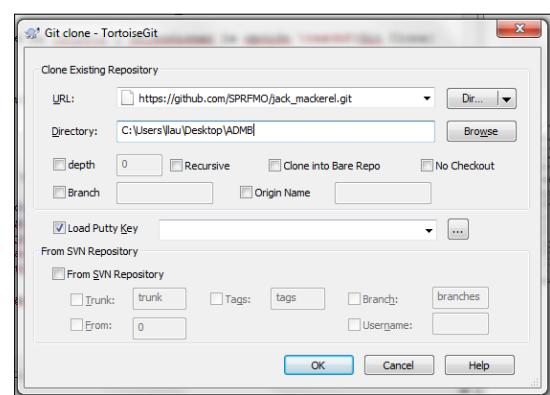


Figura 17: Menú Tortoise

5. Por defecto, lo copiado en el paso (2) se pegará en el campo **URL:**
6. En el campo **Directory:**, asegurarse que la dirección corresponda a la carpeta que acabamos de crear. Esto es porque, por defecto, se añade a la carpeta elegida el nombre del repositorio. Por ejemplo: en nuestro caso, la carpeta destino que aparecerá es Escritorio\SPRFM0_SC\jack_mackerel y debemos cambiar a Escritorio\SPRFM0_SC.
7. Dar click en **Aceptar** y esperar a que termine la copia.



(a) Opciones por defecto



(b) Opciones modificadas

Figura 18: Copia de imagen desde GitHub

2.1.2. Sincronizar una carpeta con el repositorio

Sincronizar nuestra carpeta SPRFMO_SC con el repositorio implica: a) Si se realiza un cambio dentro de la carpeta SPRFMO_SC de JJM, poder reportar ese cambio (**Commit**), y aplicarlo al repositorio (**Push**). Así mismo, si otros usuarios realizan y aplican cambios al repositorio, poder actualizar constantemente nuestra carpeta con estos cambios (**Pull**).

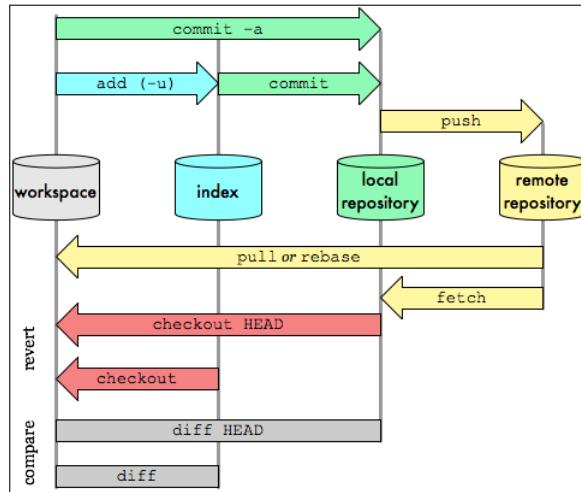


Figura 19: Comandos de Transporte de Datos Git

Para cualquier caso, lo primero será crear una cuenta en la web de GitHub desde el siguiente link: <https://github.com/join>

- Realizar un **COMMIT**: Significa consignar un cambio (o conjunto de cambios) tentativos o no permanente.
 1. Una vez realizados los cambios dentro del directorio, previamente clonado, dar click derecho en esta carpeta.
 2. Elegir la opción **Git Commit ->"master"...**

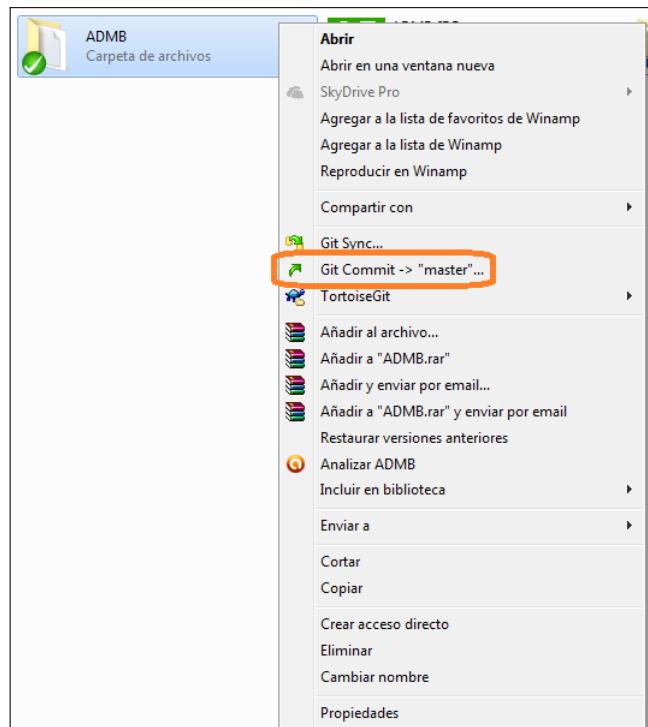


Figura 20: Comandos de Transporte de Datos Git

3. En la ventana emergente, escribir el mensaje con una descripción de cambio realizado. Este mensaje será visto en el repositorio y -por lo tanto- es necesario siempre incluirlo.

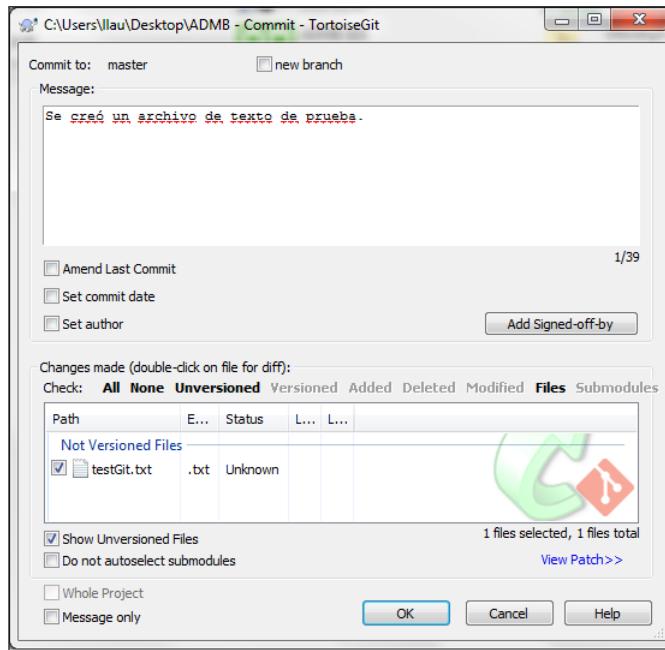
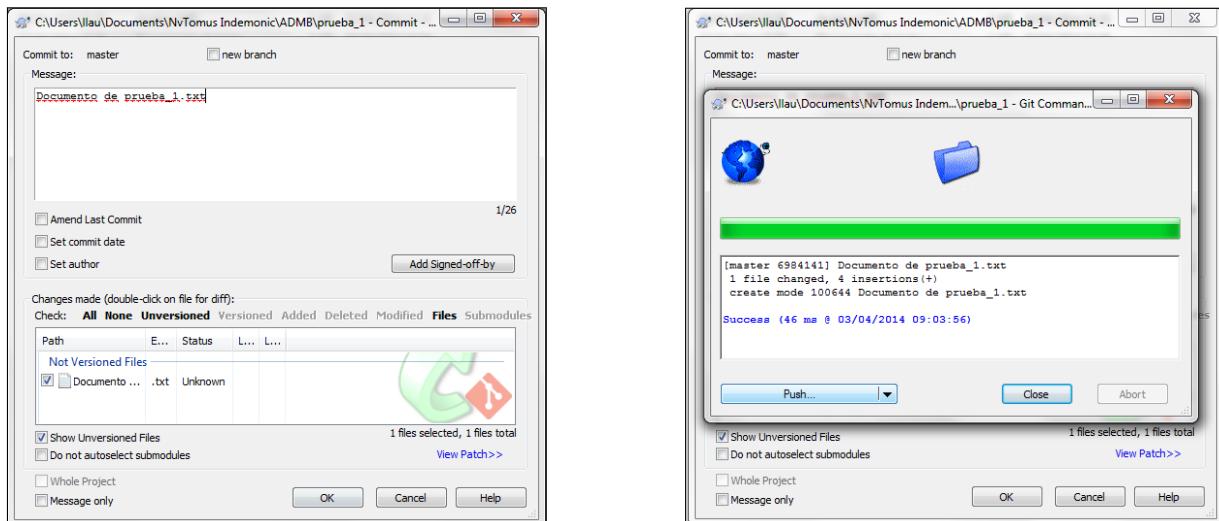


Figura 21: Comandos de Transporte de Datos Git

- Realizar un **PUSH**: Significa mover un cambio (o conjunto de cambios) a una rama del repositorio principal.

1. Una vez realizado el **Commit**, en la ventana emergente, dar click en el botón **Push**.



(a) Ventana de inicio de Commit

(b) Ventana de finalización de Commit

Figura 22: Ventanas de Commit

2. En la ventana emergente, dar click en **OK**.

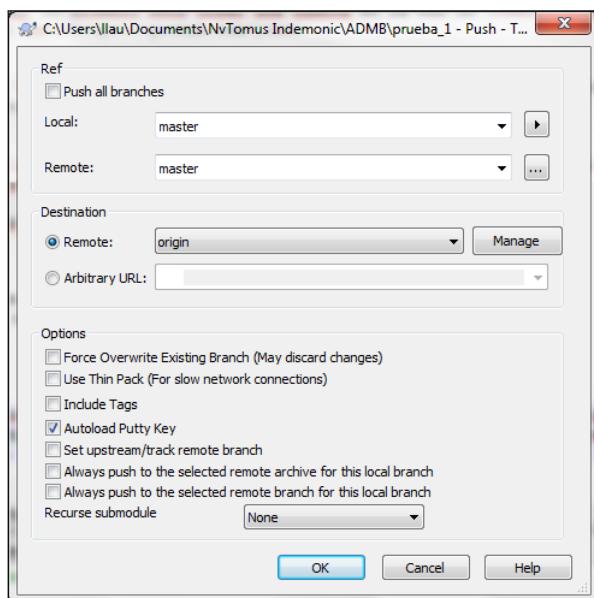


Figura 23: Ventana de Push

3. Ingresar el **Usuario** y **Password** del repositorio.

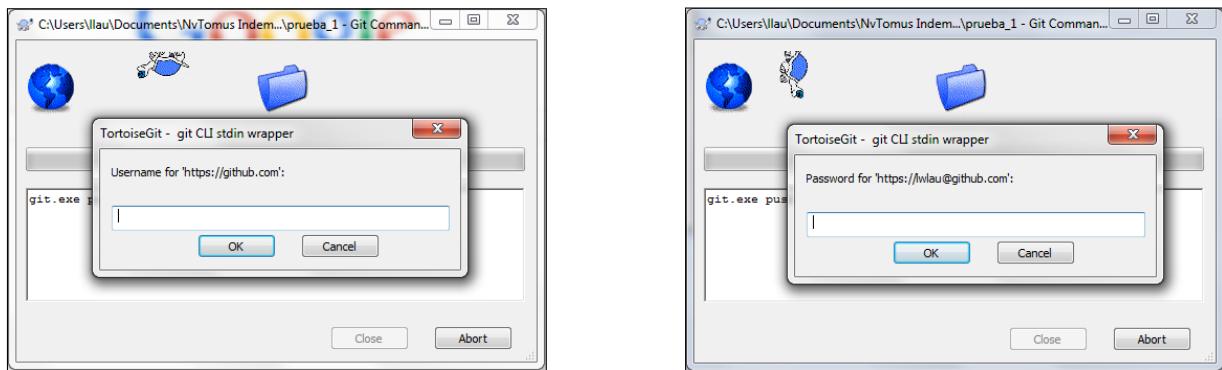


Figura 24: Ventanas de confirmación de Push

4. En la ventana final, click en **Close**.

- Realizar un **PULL**: Significa recibir los últimos cambios realizados al repositorio principal. En otros términos, es la actualización del directorio local (en nuestro ordenador) a partir del repositorio principal (nube).

1. Ir al directorio local (**SPRFM0_SC**) y dar click derecho.
2. En el menú contextual, ir al menú lista de **Tortoise** y elegir la opción **Pull**.

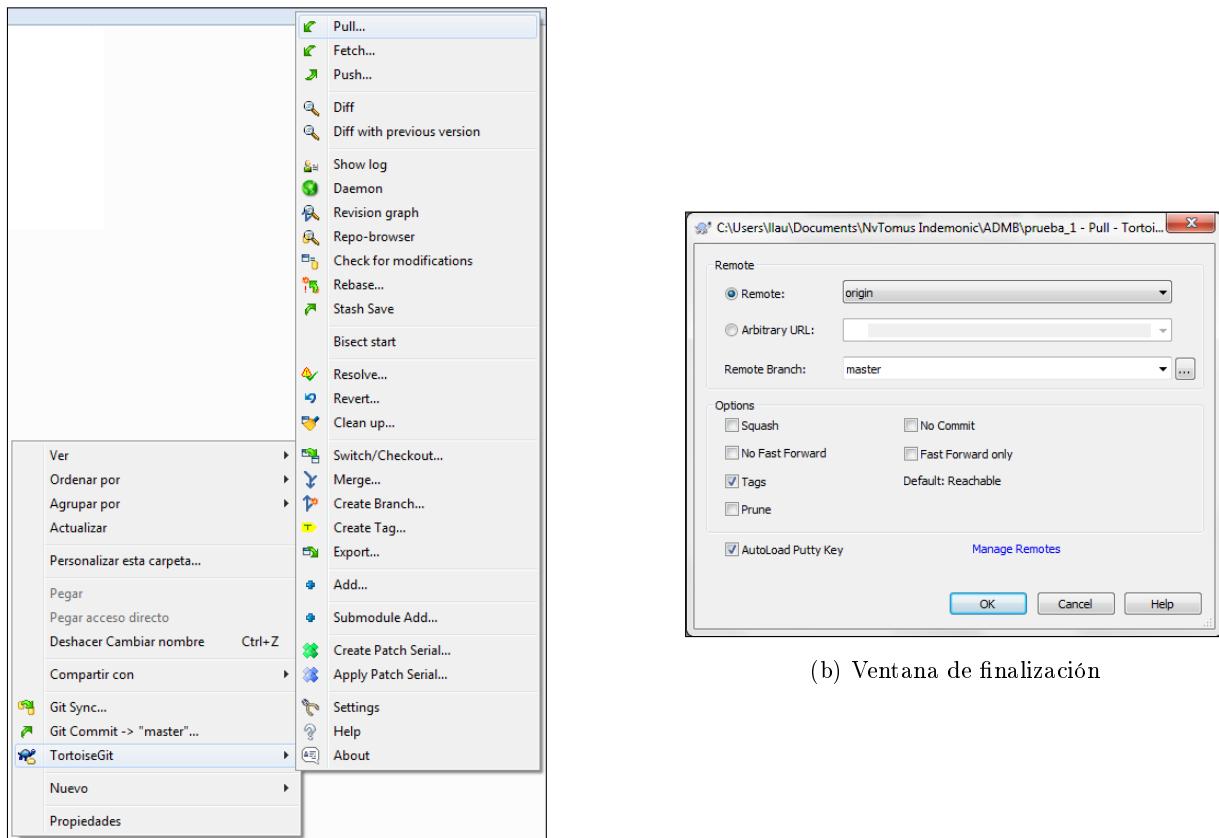


Figura 25: Ventanas de **Pull**

2.2. Compilación del modelo

C++ es el lenguaje en el que se encuentra codificado el modelo propiamente dicho. Compilar significa traducir los comandos escritos en C++ a binario (lenguaje de máquina). Para lograr este paso se requiere únicamente el archivo `jjm.tpl`, que es el que contiene al código de **JJM** para ser ejecutado en **ADMB**.

A continuación, se mostrarán los pasos para la compilación del modelo desde el editor de texto **EMACS** y desde una venta de símbolo de sistema. Se usarán los scripts de la carpeta `/Code/admb/`.

2.2.1. Compilación desde EMACS

1. Asegurarse de tener en la carpeta `admb` los siguientes archivos:

- a) `cleanad.bat`
- b) `jjm.tpl`
- c) `[modelo(s)].ctl`
- d) `[data asociada al modelo(s)].dat`
- e) `MyArchive.bat`
- f) `readme.md`
- g) `run.bat`
- h) `[archivo de parámetros iniciales asociados al modelo(s)].par` (Si fuera necesario)

2. Abrir el archivo jjm.tpl en el software EMACS.

```
// Naming Conventions:
//
// GENERAL:
//   styr, endyr beginning year and ending year of model (catch data available)
//   nages      number of age groups considered
//   nysr_      number of observations available to specific data set
//
// DATA SPECIFIC:
//
//   catch_bio  Observed catch biomass
//   fsh        fishery data
//
// Define indices
//   nind      number of indices
Index values
//   nysr_ind    Number of years of index value (annual)
//   yrs_ind     Years of index value (annual)
//   obs_ind     Observed index value (annual)
//   obs_se_ind  Observed index standard errors (annual)
Age-comp values
//   nysr_ind_age Number of years index age data available
//   yrs_ind_age  Years of index age value (annual)
//   oac_ind      Observed age comp from index
//   n_sample_ind_age Observed age comp sample sizes from index
//
//   eac_ind      Expected age comp from index
//
//   sel_ind      selectivity for egg production index
//
//   pred_ind ...
//
//   oac_fsh     Observed age comp from index
//   obs_ind_size Observed size comp from index
//
//   pred_fsh_age Predicted age comp from fishery data
//   eac_fsh     Expected age comp for fishery data (only years where data available)
//   eac_...
//
//   pred_tmp_ind Predicted index value for trawl index
//
//   sel_fsh     selectivity for fishery
//
//   sel_ch      indicates time-varying selectivity change
//
// Add bit for historical F
// Added length part for selectivity
//
```

Figura 26: Código del modelo en el entorno de EMACS

3. Presionar el botón Build y esperar a que finalice la ejecución.

```
-- mode: compilation; default-directory: "c:/Users/lau/Documents/NvTomus Indemonic/ADMB/tpl" --
Compilation started at Wed Mar 26 15:27:55
admb jhm
*** Parsing jjm.tpl:
tpl2cpp jjm
xxglobal.tmp
xxhtop.tmp
header.tmp
xxalloc.tmp
xxtopm.tmp
      1 archivo(s) copiado(s).

*** Compile jjm.cpp:
g++ -c -O3 -DSAFE_ALL -D_GNUUDOS__ -Dlinux -DUSE_LAPLACE -fpermissive -I. -IC:\admb\include\

*** Linking jjm.obj:
g++ -static -o jjm.exe jjm.obj C:\admb\contrib\lib\libcontrib.a C:\admb\lib\libadmb.a
Successfully built 'jjm.exe'.

Compilation finished at Wed Mar 26 15:28:04
--(Unix)--- jjm.tpl Top (1,0) (ADMB)-----*-compilation* All (1,0) (Compilation:exit [0])-----
```

Figura 27: Compilar el código

3. Ejecución de JJM

Una vez finalizada la compilación, se procederá a ejecutar el modelo utilizando alguna de las bases de datos disponibles en el repositorio.

3.1. Archivos necesarios

Es importante constatar -previamente- la existencia de los archivos **[nombre de modelo].ctl** y **[data asociada al modelo].dat**.

- **.ctl**: Es la extensión de archivo de control para un determinado modelo. Posee los parámetros que se usarán en la ejecución de un modelo elegido.
- **.dat**: Es la extensión de archivo que contiene los datos que el modelo utilizará al ejecutarse.

Así mismo, se debe revisar la existencia de los siguientes tres scripts (batch):

- **run.bat**: Este script contiene la órdenes para la ejecución de un determinado modelo.
- **cleanad.bat**: Este archivo limpia (elimina) archivos temporales creados en la ejecución de un determinado modelo por el script anterior.
- **MyArchive.bat**: Este script se ejecuta en conjunto con **run.bat**, y su función es crear las carpetas necesarias para la ejecución del modelo.

Para el correcto funcionamiento de los scripts de Resultados (en R), se llevarán a cabo modificaciones en los archivos **run.bat** y **cleanad.bat**:

- En el archivo **run.bat**, el código debe quedar de la siguiente manera (modificaciones en las líneas 1, 2 y 3):

```
1 :: jjm -nox -ind %1.ctl -ainp arc\%1.par -iprint 100
2 jjm -nox -ind %1.ctl -iprint 100
3 :: copy jjm.par arc\%1.par
4 copy jjm.rep arc\%1.rep
5 copy jjm.std arc\%1.std
6 copy %1.prj arc\%1.prj
7 :: copy jjm.cor arc\%1.cor
8 copy fprof.yld arc\%1.yld
9 copy for_r.rep arc\%1_R.rep
10 call cleanad.ba
```

- En el archivo **cleanad.bat**, el código debe quedar de la siguiente manera (modificación en la línea 26):

```

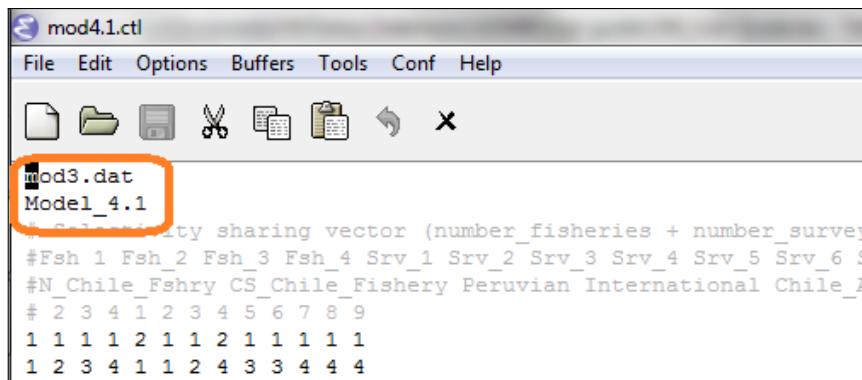
1 @echo off
2 echo cleaning admb files
3 del varssave.* >NUL
4 del cmpdiff.* >NUL
5 del tmp_admb >NUL
6 del gradfil2.* >NUL
7 del *.tmp >NUL
8 del *~ >NUL
9 del *.dep >NUL
10 del *.log >NUL
11 del *.obj >NUL
12 del *.o >NUL
13 rem del *.cpp >NUL
14 del *.htp >NUL
15 del *.0? >NUL
16 del *.r1? >NUL
17 del *.p1? >NUL
18 del *.b1? >NUL
19 del *.r0? >NUL
20 del *.p0? >NUL
21 del *.b0? >NUL
22 del *.hes >NUL
23 del variance. >NUL
24 del *.cov >NUL
25 del *.rpt >NUL
26 :: del *.cor >NUL
27 del *.eva >NUL
28 del *.td2 >NUL
29 del *.tds >NUL
30 del *.tr2 >NUL
31 echo on

```

3.2. Ejecución

A continuación, se detallarán los pasos para correr cualquiera de los modelos dentro del repositorio. Para ello se requiere conocer el nombre del archivo con extensión .ctl (archivo de control) y su respectivo .dat (data asociada).

Para averiguar el archivo "[data asociada al modelo].dat" basta con abrir el archivo [nombre de modelo].ctl con **EMACS** y verificar la primera línea.



```
mod4.1.ctl
File Edit Options Buffers Tools Conf Help
mod3.dat
Model_4.1
# Selectivity sharing vector (number_fisheries + number_survey
#Fsh_1 Fsh_2 Fsh_3 Fsh_4 Srv_1 Srv_2 Srv_3 Srv_4 Srv_5 Srv_6 S
#N Chile_Fshry CS_Chile_Fishery Peruvian International Chile_A
# 2 3 4 1 2 3 4 5 6 7 8 9
1 1 1 1 2 1 1 2 1 1 1 1 1
1 2 3 4 1 1 2 4 3 3 4 4 4
```

Figura 28: Obtención del archivo .dat relacionado

1. Abrir una ventana de comandos en la carpeta /Code/admb/

a) Abrir una ventana de explorador e ir a la carpeta anterior a admb

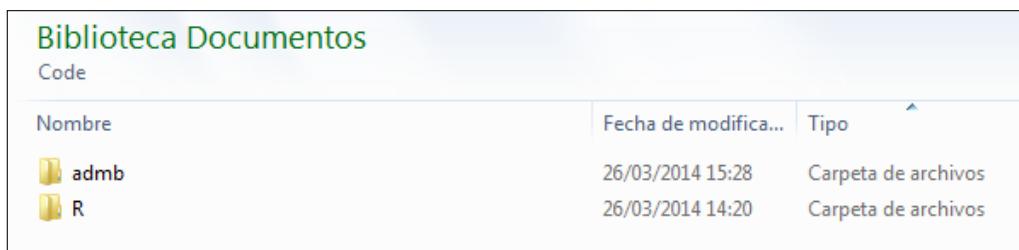


Figura 29: Carpeta Code/

b) Presionar, a la vez, Shift + Click derecho en la carpeta admb y seleccionar la opción Abrir una ventana de comandos aquí

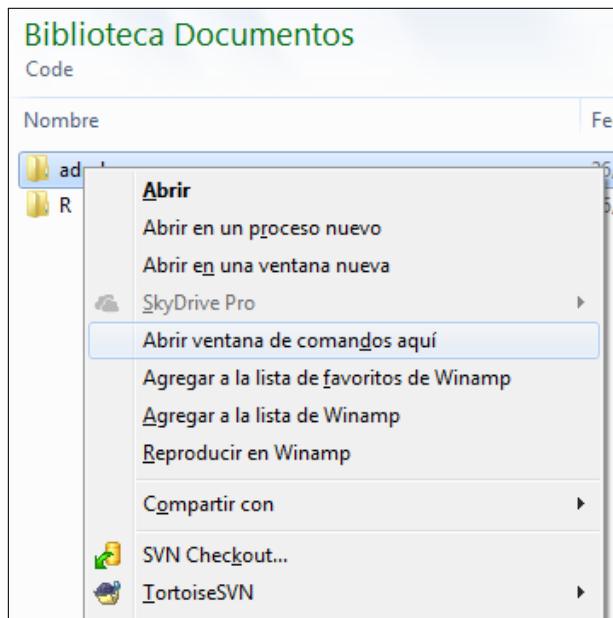


Figura 30: Opción Abrir ventana de comandos aquí

2. En la ventana de comandos abierta, ejecutar el modelo escribiendo `run [modelo]`. Se debe colocar el nombre del archivo SIN la extensión. Por ejemplo: Para correr el mod4.2, el comando será `mod4.2` y NO `mod4.2.ctl`.



Figura 31: Ventana de comandos

3. Esperar a que finalice el proceso. Mientras se ejecuta el modelo, se crearán archivos temporales que finalmente será borrados por el script **cleanad.bat**.
4. Al final del proceso se habrán creado archivos que se usarán en la generación de tablas y gráficos. De esta forma, se tiene:
 - Dentro de la carpeta **admb**: For_R.rep, Fprof.yld, jjm.cor, jjm.std, [modelo].prj, proj.dat, jjm.bar, jjm.par, jjm.rep, mceval.dat, jjm.exe, jjm.cpp
 - Dentro de la carpeta **admb/arc**: [modelo].cor, [modelo].par, [modelo].prj, [modelo].rep, [modelo].std, [modelo].yld, [modelo]_R.rep

4. Visualización de resultados

En esta sección se usarán los scripts de la carpeta `/Code/R/`. Estos han sido codificados en lenguaje R por lo que se utilizará el programa **RStudio**.

4.1. Descripción de los scripts de R

Los scripts dentro de la carpeta `/Code/R/` pueden agruparse de la siguiente manera:

- **Funciones:** Son subrutinas o subprogramas que forman parte de un algoritmo (o algoritmos) principal. Para el caso de **ADMB** lo constituyen:

- `ADMB2R_15102012.r`
- `ADMB2R_15102013.r`
- `compareRuns.r`
- `diagnostics.r`
- `diagnostics_v2.r`

- **Algoritmos principales:** Son aquellos que harán uso de las funciones (antes mencionadas) para generar resultados (cuadros y tablas). Para el caso de **ADMB** lo constituyen:

- `JJM_Assessment.r`: Genera las salidas relacionadas a cada modelo ejecutado por **JJM**. Así mismo, realiza comparaciones entre los diversos modelos ejecutados.
- `JJM_projections.r`: Realiza proyecciones en base a los modelos ejecutados y genera los gráficos correspondientes.
- `JJM_tables.r`: Genera tablas de verosimilitud en base a modelos seleccionados.

Existe, por otro lado, una carpeta llamada **HCRFramework**. Dicha carpeta contiene funciones que generan objetos tipo binario de R.

4.2. Ejecución de los scripts

4.2.1. Pasos previos

1. Verificar que la versión de R utilizada sea igual o mayor a la 3.00.

a) Click en el menú Tools y luego en la opción Global Options

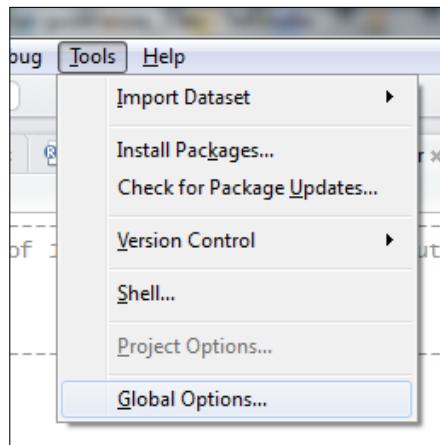


Figura 32: Llegar al menú de opciones generales en **RStudio**

b) En la pestaña General, click en el botón Change

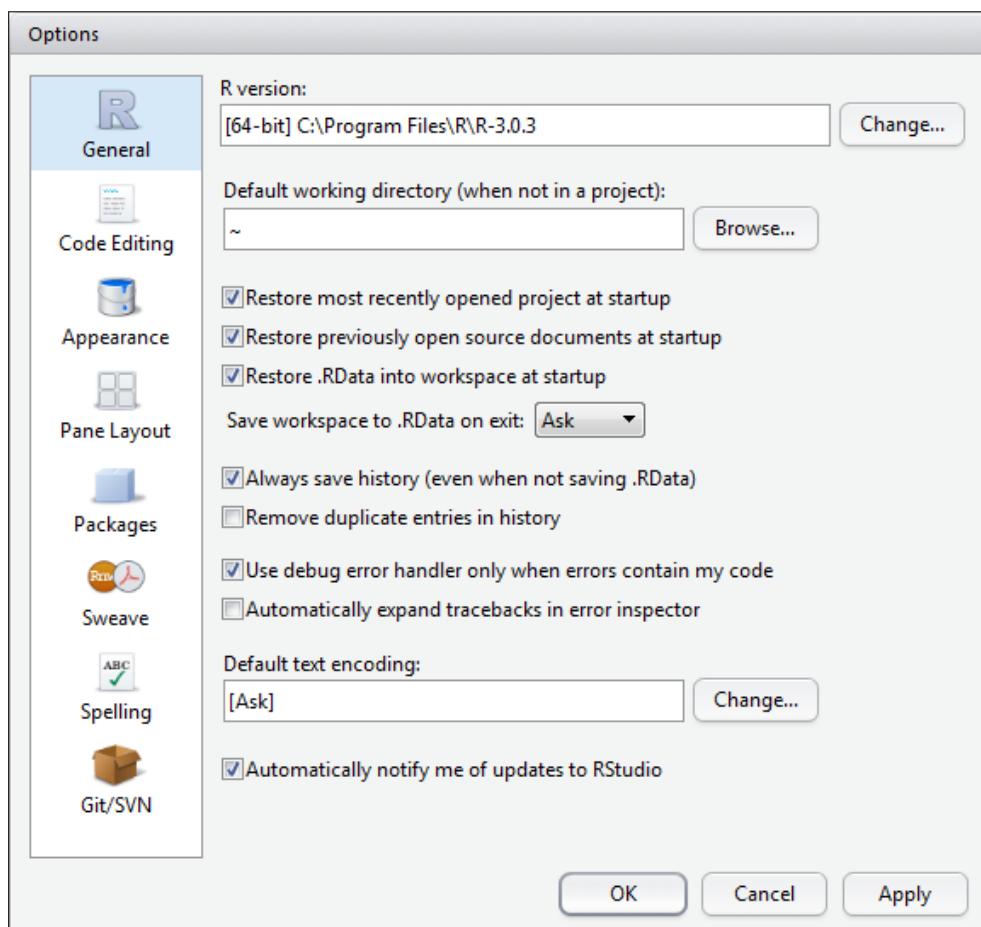


Figura 33: Opciones generales en **RStudio**

- c) Seleccionar **Choose a specific version of R:** y seleccionar la una versión igual o superior a 3.0.0 y **Aceptar.**

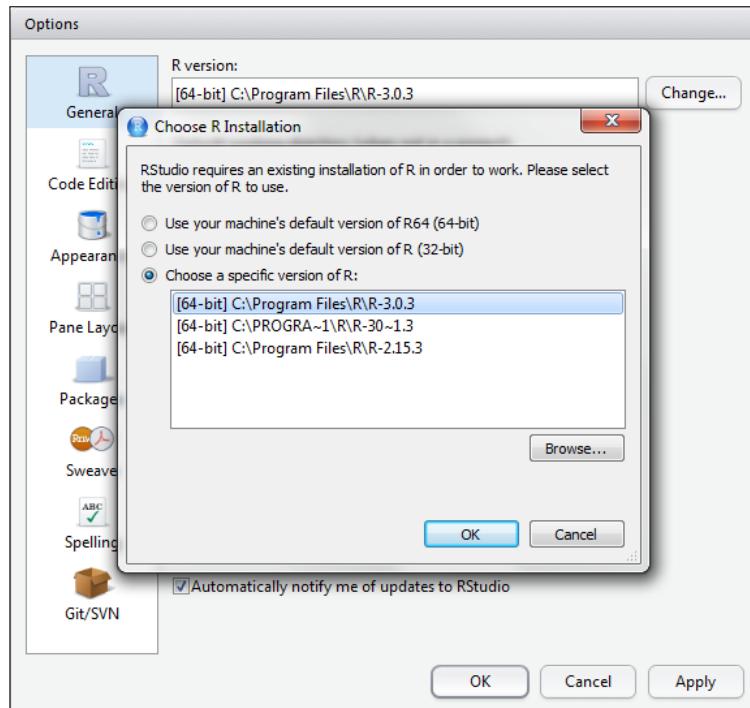


Figura 34: Seleccionar la versión a utilizar de **R**

2. Abrir RStudio e instalar los paquetes **lattice**, **PBSadmb**, **RColorBrewer** y **doBy** mediante el comando `install.packages("[nombre del paquete]")` en **Consola**.

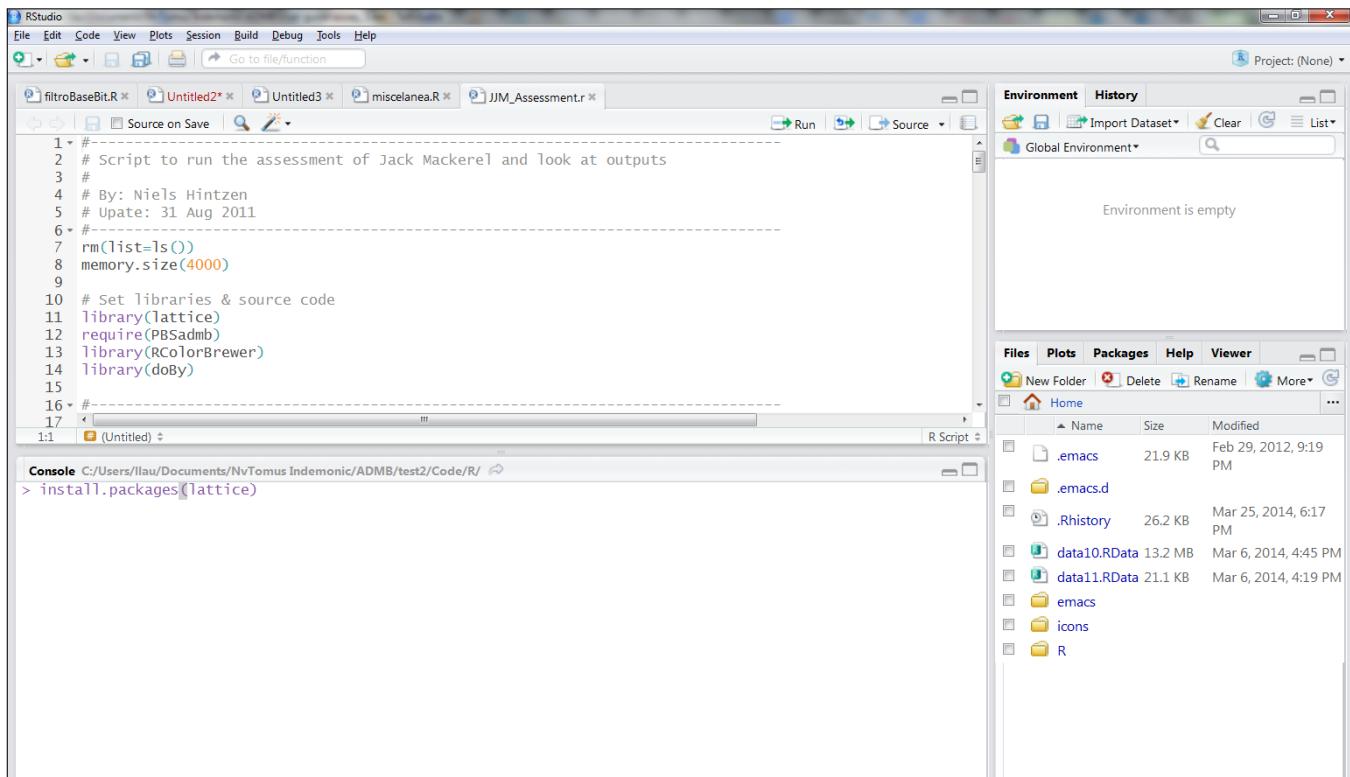


Figura 35: Ventana de **RStudio**

4.2.2. Segundo cada script

1. Abrir el archivo `JJM_Assessment.r` y modificar y correr las siguientes líneas:

```
1 # -----
2 # Set paths
3 # -----
4 reposDir    <- "[Nombre de carpeta del repositorio]"
5 codePath    <- file.path(reposDir,"Code/R/")
6 inputPath   <- file.path(reposDir,"Code/admb/")
7 outputPath  <- file.path(reposDir,"Code/admb/arc/")
8 resultPath <- file.path(reposDir,"Results/Assessment/")
9 setwd(codePath)
10
11 # Specify control file
12 controlFile <- "[nombre de modelo].ctl"
```

NOTA 1: Para correr se deben seleccionar las líneas de código deseadas y pulsar **Ctrl + ENTER**

NOTA 2: Dentro del código, los textos [entre corchetes] indican que pueden (deben) ser modificados según las necesidades del usuario y/o características de su entorno (Sistema Operativo).

NOTA 3: El símbolo `#` precede a un comentario dentro del código. Es decir, un texto que el programa no tomará en cuenta.

NOTA 4: Las salidas se escribirán en formato pdf en la carpeta indicada en la línea:

```
resultPath <- file.path(reposDir,"Results/Assessment/")
```

2. Dentro del script `Assessment.r` se pueden obtener resultados por modelo y de comparación entre modelos.

a) Para la sección **Assessment**, modificar y correr las siguientes líneas de código. Estas líneas procesarán y escribirán salidas para cada modelo indicado. En el ejemplo que se muestra a continuación, se detalla el código usado para obtener resultados del modelo `mod4.3`.

```
30 # Run the assessment
31 source("diagnostics_v2.r")
32 source("ADMB2R_15102013.r")
33 # system(paste("../admb/jjm.exe",' -ind',paste(controlFile,'.ctr',sep=""),'-nox'), wait =
      TRUE)
34
35 # Read in the output of the assessment
36 run_name="mod4.3"
37 dat_name="mod3.dat"
38
39 jjm.in <- read.dat(iFilename = paste(dat_name,sep=""),iPath=inputPath)
40 jjm.out <- readList(file.path(inputPath,paste("arc/",run_name,".r.rep",sep="")))
41 jjm.ypr <- readyPR(file.path(inputPath,paste( "arc/",run_name,".yld", sep="")))
42
43 # -----
44 # Create diagnostics
45 # -----
46 pdf(paste(resultPath,"summary_",run_name,".pdf",sep=""),height=29.7/2.54,
      width=21/2.54,pointsize = 16, bg = "white")
48 diagnostics(jjm.out,jjm.in,jjm.ypr,what=c("input","fit","projections","ypr"))
49 dev.off()
50
51 #Write output to file
52 writeList(setOutputNames(jjm.out),fname=paste(controlFile,"_out.txt",sep=""),format="P")
```

b) Si se desea obtener resultados de comparación entre modelos, ir a la sección (en el código) precedida por el encabezado **Visual compare runs**.

c) Observar las siguientes líneas que se modificarán de acuerdo al número y nombre de modelos que se desea comparar.

```
59 jjm0.0 <- readList(file.path(inputPath,paste("arc/Mod0.0_r.rep",sep="")))
60 jjm0.1 <- readList(file.path(inputPath,paste("arc/Mod0.1_r.rep",sep="")))
61 jjm0.2 <- readList(file.path(inputPath,paste("arc/Mod0.2_r.rep",sep="")))
62 jjm0.3 <- readList(file.path(inputPath,paste("arc/Mod0.3_r.rep",sep="")))
63 jjm0.4 <- readList(file.path(inputPath,paste("arc/Mod0.4_r.rep",sep="")))
64 jjm0.5 <- readList(file.path(inputPath,paste("arc/Mod0.5_r.rep",sep="")))
65 jjm0.6 <- readList(file.path(inputPath,paste("arc/Mod0.6_r.rep",sep=")))
66
67 lstOuts <- list(
68   Model_0.0= jjm0.0,
69   Model_0.1= jjm0.1,
70   Model_0.2= jjm0.2,
71   Model_0.3= jjm0.3,
72   Model_0.4= jjm0.4,
73   Model_0.5= jjm0.5
74     )
75 lstOuts <- list(
76   Model_0.5= jjm0.5,
77   Model_0.6= jjm0.6
78 )
79 pdf(paste(outputPath,"Compare_1_4.pdf",sep=""),height=29.7/2.54,width=21/2.54,pointsize = 24, bg = "white")
```

d) Las líneas del siguiente tipo se modificarán -en los textos resaltados- de la siguiente manera:

```
jjm0.0 <- readList(file.path(inputPath,paste(.arc/Mod0.0_r.rep",sep=)))
```

- 1) **m0.0**: Se colocará el número correspondiente al modelo que se desea comparar. Por ejemplo **m4.1**
- 2) **Mod0.0**: Se colocará el nombre del archivo de extensión **.rep** correspondiente al modelo que se desea comparar. Por ejemplo **mod4.1**. Se debe verificar, previamente, la existencia de este archivo en la carpeta **/Code/admb/arc/**.

e) Las líneas del siguiente tipo se modificarán -en los textos resaltados- de la siguiente manera:

```
Model_0.0 = jjm0.0,
```

- 1) **Model_0.0**: Se colocará el número correspondiente al modelo que se desea comparar. Por ejemplo **Model_4.1**
- 2) **jjm0.0**: Se colocará el nombre utilizado en el paso anterior (línea anterior). Es decir, si se utilizó el nombre de **jjm0.0 <- ...**, se debe usar el mismo en esta parte.

NOTA 5: Crear tantas líneas según el número de modelos que se desea comparar. Recordar que los textos [entre corchetes] se modifican según corresponda.

f) Modificar en el texto resaltado de la siguiente línea:

```
pdf(paste(outputPath,"Compare_1_4.pdf",sep="),height=29.7/2.54,width=21/2.54,pointsize = 24, bg = "white")
```

g) Colocar el nombre del archivo de salida

h) Correr las líneas indicadas. El archivo de salida se guardará en la carpeta **/Code/admb/arc/**.