

Instalación y Configuración Autónoma de RStan

Capacitación

2025-10-02

Contents

Introducción	1
1 Instalar R	1
2 Instalar RStudio Desktop	2
3 Instalar RTools (Solo para Windows)	2
4 Detección del sistema operativo	2
5 Instalación de librerías necesarias	2
6 Configuración de RStan	3
6.1 Windows: Verificación e instalación de RTools	3
6.2 macOS y Linux	3
7 Verificación de RStan	3

Introducción

Bienvenido a la guía de instalación. Este manual ha sido diseñado para simplificar el proceso de configuración de un entorno de trabajo completo para el taller de *Benchmarking de cadenas MCMC* para la estimación de indicadores del mercado laboral, específicamente en el contexto de la *Estimación de Áreas Pequeñas (SAE)*.

La guía te llevará paso a paso a través de la instalación de los componentes esenciales, incluyendo:

- **R y RStudio:** El lenguaje de programación y su entorno de desarrollo integrado (IDE).
- **RTools:** Un conjunto de herramientas cruciales para la compilación de paquetes en Windows.
- **Stan y RStan:** La plataforma de modelado probabilístico y su interfaz con R, herramientas poderosas para la inferencia bayesiana.

1 Instalar R

Primero, debes instalar el lenguaje de programación R. Es la base sobre la cual funciona RStudio.

- Ve a la página de descarga oficial de **R** en CRAN (Comprehensive R Archive Network).
- Selecciona tu sistema operativo (Windows, macOS o Linux).
- Haz clic en “**base**” y luego descarga la versión más reciente del instalador.
- Ejecuta el archivo descargado y sigue las instrucciones del asistente de instalación. Puedes dejar la mayoría de las opciones por defecto.

2 Instalar RStudio Desktop

Una vez que tengas R instalado, puedes instalar RStudio, el entorno de desarrollo recomendado para trabajar con R.

- Ve a la página de descarga de **RStudio** en Posit.
- Desplázate hacia abajo hasta la sección de descargas y elige la versión gratuita de **RStudio Desktop**.
- El sitio web detectará automáticamente tu sistema operativo y te recomendará el instalador correcto. Haz clic en el botón de descarga.
- Ejecuta el archivo descargado y sigue las instrucciones del asistente de instalación.

3 Instalar RTools (Solo para Windows)

RTools es un conjunto de herramientas de desarrollo necesario en Windows para que R compile paquetes desde su código fuente. Esto es crucial para la instalación de librerías como **RStan**.

- Ve a la página de **RTools** en CRAN.
- Descarga la versión de RTools que corresponda a tu versión de R (por ejemplo, R 4.3 o superior).
- Ejecuta el instalador. Es muy importante que, durante el proceso, marques la opción “**Add Rtools to system PATH**” para que RStudio pueda encontrarlo automáticamente.
- Una vez que la instalación esté completa, puedes verificar que RTools está funcionando correctamente abriendo RStudio y ejecutando el siguiente comando en la consola:

```
pkgbuild::has_build_tools(debug = TRUE)
```

El resultado debería ser ``TRUE``, indicando que las herramientas de compilación están listas.

4 Detección del sistema operativo

```
os <- Sys.info()["sysname"]  
message("Sistema operativo detectado: ", os)
```

5 Instalación de librerías necesarias

```
# Lista de paquetes requeridos  
paquetes <- c("bayesplot", "fastDummies", "haven", "kableExtra",  
  "knitr", "magrittr", "parallel", "patchwork", "posterior", "printr",  
  "rstan", "rstanarm", "sampling", "srvyr", "survey", "tibble",  
  "tidyverse"  
)  
  
# Instalar paquetes faltantes automáticamente  
instalar <- paquetes[!paquetes %in% installed.packages()[,"Package"]]  
if(length(instalar) > 0){  
  install.packages(instalar, dependencies = TRUE)  
} else {  
  message("Todos los paquetes ya están instalados.")  
}
```

6 Configuración de RStan

6.1 Windows: Verificación e instalación de RTools

```
if(!pkgbuild::has_build_tools(debug = TRUE)){  
  message("RTools no encontrado. Descárguelo desde: https://cran.r-project.org/bin/windows/Rtools/")  
} else {  
  message("RTools detectado correctamente.")  
}  
  
# Configuración del compilador C++ para RStan  
dotR <- file.path(Sys.getenv("HOME"), ".R")  
if(!dir.exists(dotR)) dir.create(dotR)  
makevars <- file.path(dotR, "Makevars.win")  
if(!file.exists(makevars)) file.create(makevars)  
cat("\nCXX14FLAGS=-O3 -march=native -mtune=native",  
    file = makevars, sep = "\n", append = TRUE)  
message("Configuración de compilador completada.")
```

6.2 macOS y Linux

```
message("En macOS o Linux, asegúrese de tener un compilador C++ adecuado.")  
# macOS: Xcode Command Line Tools  
# Linux: build-essential (Ubuntu/Debian)
```

7 Verificación de RStan

```
library(rstan)  
options(mc.cores = parallel::detectCores())  
  
stan_code <- '  
data {  
  int<lower=0> N;  
  int<lower=0,upper=1> y[N];  
}  
parameters {  
  real<lower=0,upper=1> theta;  
}  
model {  
  theta ~ beta(1,1);  
  y ~ bernoulli(theta);  
}  
'  
  
data <- list(N=10, y=c(0,1,0,1,0,1,0,1,0,1))  
fit <- stan(model_code=stan_code, data=data, iter=500, chains=2)  
saveRDS(fit, file = "fit.rds")  
  
fit <- readRDS("fit.rds")  
print(fit)
```

```
## Inference for Stan model: anon_model.  
## 2 chains, each with iter=500; warmup=250; thin=1;
```

```

## post-warmup draws per chain=250, total post-warmup draws=500.
##
##      mean se_mean   sd  2.5%  25%  50%  75% 97.5% n_eff Rhat
## theta  0.50    0.01 0.13   0.24  0.41  0.50  0.59  0.77  147 1.01
## lp__  -8.81    0.06 0.81 -11.81 -8.88 -8.51 -8.36 -8.32  157 1.00
##
## Samples were drawn using NUTS(diag_e) at Mon Sep 29 10:55:14 2025.
## For each parameter, n_eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor on split chains (at
## convergence, Rhat=1).

```