Ciencia de datos para el análisis político y desarrollo de políticas públicas con R

Jeshua Romero Guadarrama

2021-10-18

# Prefacio

Publicado por Jeshua Romero Guadarrama en colaboración con JeshuaNomics:

Git Hub Facebook Twitter Linkedin Vkontakte Tumblr YouTube Instagram

Jeshua Romero Guadarrama es economista y actuario por la Universidad Nacional Autónoma de México, quien ha construido el presente proyecto en colaboración con JeshuaNomics, ubicado en la Ciudad de México, se puede contactar mediante el siguiente correo electrónico: [jeshuanomics@gmail.com](mailto:jeshuanomics@gmail.com). Última actualización el lunes 18 del 10 de 2021

Los estudiantes con poca experiencia en el análisis avanzado de políticas a menudo tienen dificultades para entender los beneficios de desarrollar habilidades de programación estadística en **R** al momento de aplicar diversos métodos descriptivos e inferenciales. Análisis político con R por Jeshua Romero Guadarrama (2021), ofrece una introducción interactiva a los aspectos esenciales de la programación por medio del lenguaje y software estadístico **R**, así como una guía para la aplicación de la teoría política y análisis detallado de políticas públicas en entornos específicos. En otras palabras, el objetivo es que los estudiantes se adentren al mundo de la política aplicada mediante ejemplos empíricos presentados en la vida diaria y haciendo uso de las habilidades de programación recién adquiridas. Dicho objetivo se encuentra respaldado por ejercicios de programación interactivos y la incorporación de visualizaciones dinámicas de conceptos fundamentales mediante la flexibilidad de JavaScript, a través de la biblioteca D3.js.

En los últimos años, el lenguaje de programación estadística **R** se ha convertido en una parte integral del plan de estudios de las clases de análisis político y estadística que se imparten en las universidades. Regularmente una gran parte de los estudiantes no han estado expuestos a ningún lenguaje de programación antes y, por lo tanto, tienen dificultades para participar en el aprendizaje de **R** por sí mismos. Con poca experiencia en el análisis avanzado de estadísticas, es natural que los novicios tengan dificultades para comprender los beneficios de desarrollar habilidades en **R** para aprender y aplicar la estadística. Estos incluyen particularmente la capacidad de realizar, documentar y comunicar estudios empíricos y tener las facilidades para programar estudios de simulación, lo cual es útil para, por ejemplo, comprender y validar teoremas que generalmente no se asimilan o entienden fácilmente con el estudio de las fórmulas. Al ser un economista aplicado y analista político, me gustaría que mis colegas desarrollen capacidades de gran valor; en consecuencia, deseo compartir con las nuevas generaciones de politólogos y economistas mis conocimientos.

En lugar de confrontar a los estudiantes con ejercicios de codificación puros y literatura clásica complementaria, he pensado que sería mejor proporcionar material de aprendizaje interactivo que combine el código en **R** con el contenido del curso de texto *Introducción a la Econometría* de [Stock and Watson](#ref-stock2015) ([2015](#ref-stock2015)) que sirve de base para el presente material. El presente trabajo es un complemento empírico interactivo al estilo de un informe de investigación reproducible que permite a los estudiantes no solo aprender cómo los resultados de los estudios de casos se pueden replicar con **R**, sino que también fortalece su capacidad para utilizar las habilidades recién adquiridas en otras aplicaciones empíricas.

#### Las convenciones usadas en el presente curso

* El texto *en cursiva* indica nuevos términos, nombres, botones y similares.
* El texto **en negrita** se usa generalmente en párrafos para referirse al código **R**. Esto incluye comandos, variables, funciones, tipos de datos, bases de datos y nombres de archivos.
* Texto de ancho constante sobre fondo gris indica un código **R** que usted puede escribir literalmente. Puede aparecer en párrafos para una mejor distinción entre declaraciones de código ejecutables y no ejecutables, pero se encontrará principalmente en forma de grandes bloques de código **R**. Estos bloques se denominan fragmentos de código.

#### Reconocimiento

A mi alma máter: Universidad Nacional Autónoma de México (Facultad de Economía y Facultad de Ciencias). Por brindarme valiosas oportunidades que coadyuvaron a mi formación.

# Índice de contenido

1. Obtención de R y descarga de paquetes

* Antecedentes e instalación
  + ¿Dónde puedo conseguir R?
* Primeros pasos: una primera sesión en R
* Ahorro de entrada y salida
* Gestión de sesiones de trabajo
* Recursos
* Problemas de práctica

1. Carga y manipulación de datos

* Lectura de datos
  + Lectura de datos de otros programas
  + Tramas de datos en R
  + Escritura de datos
* Visualización de atributos de los datos
* Declaraciones lógicas y generación de variables
* Datos de limpieza
  + Subconjuntos de datos
  + Recodificación de variables
* Fusionar y remodelar datos
* Problemas de práctica

1. Visualización de datos

* Gráficos univariados en el paquete base
  + Gráficos de barras
* La función de la trama
  + Gráficos lineales con gráfico
  + Figura Construcción con parcela: Detalles adicionales
  + Funciones complementarias
* Usando gráficos de celosía en R
* Salida gráfica
* Problemas de práctica

1. Estadísticas descriptivas

* Medidas de tendencia central
  + Tablas de frecuencia
* Medidas de dispersión
  + Cuantiles y percentiles
* Problemas de práctica

1. Inferencias básicas y asociación bivariante

* Pruebas de significancia para medias
  + Prueba de diferencia de medias de dos muestras, muestras independientes
  + Comparación de medias con muestras dependientes
* Tabulaciones cruzadas
* Coeficientes de correlación
* Problemas de práctica

1. Modelos lineales y diagnósticos de regresión

* Estimación con mínimos cuadrados ordinarios
* Diagnóstico de regresión
  + Forma funcional
  + Heteroscedasticidad
  + Normalidad
  + Multicolinealidad
  + Valores atípicos, apalancamiento y puntos de datos influyentes
* Problemas de práctica

1. Modelos lineales generalizados

* Resultados binarios
  + Modelos Logit
  + Modelos Probit
  + Interpretación de resultados logit y probit
* Resultados ordinales
* Recuentos de eventos
  + Regresión de Poisson
  + Regresión binomial negativa
  + Trazado de recuentos previstos
* Problemas de práctica

1. Uso de paquetes para aplicar modelos avanzados

* Modelos multinivel con lme4
  + Regresión lineal multinivel
  + Regresión logística multinivel
* Métodos bayesianos con MCMCpack
  + Regresión lineal bayesiana
  + Regresión logística bayesiana
* Inferencia causal con cem
  + Desequilibrio de covariables, implementación de CEM y ATT
  + Explorando diferentes soluciones CEM
* Análisis de nominaciones legislativas con wnominate
* Problemas de práctica

1. Análisis de series de tiempo

* El método Box-Jenkins
  + Funciones de transferencia frente a modelos estáticos
* Extensiones a modelos de regresión lineal por mínimos cuadrados
* Autorregresión vectorial
* Más información sobre el análisis de series de tiempo
* Código de serie temporal alternativo
* Problemas de práctica

1. Álgebra lineal con aplicaciones de programación

* Creación de vectores y matrices
  + Creando Matrices
  + Conversión de matrices y marcos de datos
  + Suscripción
* Comandos vectoriales y matriciales
  + Álgebra de matrices
* Ejemplo aplicado: Programación de regresión OLS
  + Cálculo de OLS a mano
  + Escribir un estimador de MCO en R
  + Otras aplicaciones
* Problemas de práctica

1. Herramientas de programación adicionales

* Distribuciones de probabilidad
* funciones
* Bucles
* Ramificación
* Optimización y estimación de máxima verosimilitud
* Programación orientada a objetos
  + Simular un juego
* Análisis de Monte Carlo: un ejemplo aplicado
  + Función log-verosimilitud de disuasión estratégica
  + Evaluación del estimador
* Problemas de práctica

Referencias bibliográficas

# 1 Obtener R y descargar paquetes

Palabras clave

* Línea de comando
* Operador de asignación
* Codigo de entrada
* Archivo de comandos
* División entera

Este capítulo está escrito para el usuario que nunca ha descargado **R** en su computadora, y mucho menos ha abierto el programa. El capítulo ofrece algunos antecedentes breves sobre lo que es el programa y luego describe cómo se puede descargar e instalar **R** de forma totalmente gratuita. Además, el capítulo enumera algunos recursos en internet sobre **R** que los usuarios pueden desear consultar siempre que tengan preguntas que no se abordan en este libro.

## Antecedentes e instalación\*\*

R es una plataforma para el lenguaje de programación estadística orientado a objetos S. S fue desarrollado inicialmente por John Chambers en Bell Labs, mientras que R fue creado por Ross Ihaka y Robert Gentleman. R se utiliza ampliamente en estadísticas y se ha vuelto bastante popular en Ciencias Políticas durante la última década. El programa también se ha vuelto más utilizado en el mundo empresarial y en el trabajo gubernamental, por lo que la formación como usuario de R se ha convertido en una habilidad comercializable. R, que es shareware, es similar a S-plus, que es la plataforma comercial de S. Esencialmente, R se puede usar como un lenguaje de programación basado en matrices o como un paquete estadístico estándar que opera de manera muy similar a los programas vendidos comercialmente Stata. SAS y SPSS.

**Fig. 1.1** La página de inicio de la red de archivos comprensivos (CRAN). El cuadro superior, “Descargar e instalar R,” ofrece enlaces de instalación para los tres sistemas operativos principales.

## **1.2 ¿Dónde puedo conseguir *R*?**

La belleza de R es que es shareware, por lo que es gratis para cualquiera. Para obtener R para Windows, Mac o Linux, simplemente visite la completa red de archivos de R (CRAN) en [http://www.cran.r-project.org/.](http://www.cran.r-%20project.org/) La Figura 1.1 muestra la página de inicio de este sitio web. Como puede verse, en la parte superior de la página hay un recuadro con la etiqueta *Descargar e instalar R*. Dentro de este hay enlaces para la instalación usando los sistemas operativos Linux, Mac OS X y Windows. En el caso de Mac, al hacer clic en el enlace aparecerá un archivo descargable con el sufijo pkg que instalará la última versión. Para Windows, se presentará un enlace llamado base, que conduce a un archivo exe para descargar e instalar. En cada sistema operativo, abrir el archivo respectivo guiará al usuario a través del proceso de instalación automatizado[[1]](#footnote-27). En este sencillo procedimiento, un usuario puede instalar R en su máquina personal en cinco minutos.

A medida que pasan los meses y los años, los usuarios observarán el lanzamiento de nuevas versiones de R. No hay parches de actualización para R, por lo que a medida que se lanzan nuevas versiones, debe instalar completamente una nueva versión siempre que desee actualizar a la última edición. Los usuarios no necesitan reinstalar todas las versiones que se lanzan. Sin embargo, a medida que pasa el tiempo, las bibliotecas complementarias (que se describen más adelante en este capítulo) dejarán de admitir versiones anteriores de R.Una guía potencial sobre este punto es actualizar a la versión más reciente siempre que no se pueda instalar una biblioteca de interés debido a la falta. de apoyo. El único inconveniente importante que plantea la reinstalación completa es que las bibliotecas complementarias creadas por el usuario tendrán que reinstalarse, pero esto se puede hacer según sea necesario.

## **1.3 Introducción: Una primera sesión en *R***

Una vez que haya instalado R, habrá un ícono en el menú Inicio de Windows (con la opción de colocar un acceso directo en el Escritorio) o en la carpeta Aplicaciones de Mac (con la opción de mantener un ícono en el Dock del área de trabajo). Al hacer clic o hacer doble clic en el icono se iniciará R. La figura 1.2 muestra la ventana asociada con la versión Mac del software. Notará que R tiene algunas opciones de botones en la parte superior de la ventana. Dentro de Mac, la barra de menú en la parte superior del espacio de trabajo también incluirá algunos menús desplegables. En Windows, los menús desplegables también se presentarán dentro de la ventana R. Sin embargo, con solo un puñado de menús y botones, los comandos en R se ingresan principalmente a través del código de usuario. Los usuarios que deseen la opción alternativa de tener más menús y botones disponibles pueden desear instalar RStudio o un programa similar que agregue una interfaz de apuntar y hacer clic a R, pero el conocimiento de la sintaxis es esencial. La Figura 1.3 muestra la ventana asociada con la versión Mac de RStudio[[2]](#footnote-29).

Los usuarios pueden enviar su código a través de archivos de script (la opción recomendada, descrita en la Sección 1.3) o en la línea de comando que se muestra en la parte inferior de la consola R. En la Fig. 1.2, el indicador se ve así:

**Fig. 1.2** R Console para una nueva sesión

Siempre que escriba código directamente en el símbolo del sistema, si un usuario escribe un solo comando que abarca varias líneas, el símbolo del sistema se convierte en un signo más (+) para indicar que el comando no está completo. El signo más no indica ningún problema o error, solo le recuerda al usuario que el comando anterior aún no está completo. El cursor se coloca automáticamente allí para que el usuario pueda ingresar comandos.

En la edición electrónica de este libro, la sintaxis de entrada y las impresiones de salida de R estarán codificadas por colores para ayudar a distinguir lo que el usuario debe escribir en un archivo de secuencia de comandos de los resultados esperados. El código de entrada se escribirá en fuente de teletipo azul. La salida R se escribirá en fuente de teletipo negro. Los mensajes de error que devuelva R se escribirán en fuente de teletipo rojo. Estos colores corresponden a la codificación de colores que utiliza R para el texto de entrada y salida. Si bien es posible que los colores no sean visibles en la edición impresa, el texto del libro también distinguirá las entradas de las salidas. Además, los nombres de las variables se escribirán en **negrita**. Las palabras clave conceptuales de estadística y programación, así como el texto enfatizado, se escribirán en *cursiva*. Finalmente, cuando el significado de un comando no sea evidente, las iniciales identificativas estarán subrayadas y en negrita en el texto. Por ejemplo, el comando lm significa **l**inear **m**odel.

**Fig. 1.3** Ventana abierta desde una sesión de RStudio

Al escribir la sintaxis R en la línea de comando o en un archivo de script, los usuarios deben tener en cuenta algunos preliminares importantes:

* Las expresiones y comandos en R distinguen entre mayúsculas y minúsculas. Por ejemplo, la función var devuelve la varianza de una variable: una función simple discutida en el Capítulo 4. Por el contrario, el comando VAR de la biblioteca vars estima un modelo de autorregresión vectorial, una técnica avanzada que se analiza en el capítulo 9. De manera similar, si un nombre de usuario es un conjunto de datos mydata, entonces no se puede llamar con los nombres MyData, MyData, MYDATA o myData. R supondría que cada uno de estos nombres indica un significado diferente.
* Las líneas de comando no necesitan estar separadas por ningún carácter especial como un punto y coma como en Limdep, SAS o Gauss. Una simple devolución dura servirá.
* R ignora todo lo que sigue al carácter de almohadilla (#) como comentario. Esto se aplica cuando se utiliza la línea de comandos o archivos de secuencia de comandos, pero es especialmente útil al guardar notas en archivos de secuencia de comandos para su uso posterior.
* El nombre de un objeto debe comenzar con un carácter alfabético, pero puede contener caracteres numéricos a partir de entonces. Un punto también puede formar parte del nombre de un objeto. Por ejemplo, x.1 es un nombre válido para un objeto en R.
* Puede utilizar las teclas de flecha del teclado para desplazarse hacia atrás a los comandos anteriores. Una pulsación de la flecha hacia arriba recupera el comando introducido anteriormente y lo coloca en la línea de comandos. Cada pulsación adicional de la flecha se mueve a un comando anterior al que aparece en la línea de comando, mientras que la flecha hacia abajo llama al comando que sigue al que aparece en la línea de comando.

Aparte de este puñado de reglas importantes, el símbolo del sistema en R tiende a comportarse de una manera intuitiva, devolviendo respuestas a los comandos de entrada que podrían adivinarse fácilmente. Por ejemplo, en su nivel más básico, R funciona como una calculadora de gama alta. Algunos de los comandos *aritméticos* clave son: suma (+), resta (-), multiplicación ( \*), división (/), exponenciación (^), la función módulo (%%) y división entera (% / %). Los paréntesis ( ) especifican el orden de las operaciones. Por ejemplo, si escribimos la siguiente entrada:

(3+5/78)^3\*7

Luego, R imprime la siguiente salida:

[1] 201.3761

Como otro ejemplo, podríamos preguntarle a R cuál es el resto al dividir 89 entre 13 usando la función módulo:

89%%13

Entonces R proporciona la siguiente respuesta:

[1] 11

Si quisiéramos que R realizara una división de enteros, podríamos escribir:

89%/%13

Nuestra respuesta de salida a esto es:

[1] 6

El comando de opciones permite al usuario modificar los atributos de la salida. Por ejemplo, el argumento de dígitos ofrece la opción de ajustar cuántos dígitos se muestran. Esto es útil, por ejemplo, cuando se considera la precisión con la que desea presentar los resultados que sean razonables. Otras infunciones construidas útiles de una álgebra y una trigonometría incluyen: sin(x), cos(x), tan(x), exp(x), log(x), sqrt(x) y pi. Para aplicar algunas de estas funciones, primero podemos expandir el número de dígitos impresos y luego pedir el valor de la constante :

options(digits=16)  
pi

En consecuencia, R imprime el valor de hasta 16 dígitos:

[1] 3,141592653589793

También podemos usar comandos como pi para insertar el valor de dicha constante en una función. Por ejemplo, si quisiéramos calcular el seno de un ángulo en radianes (o 90°), podríamos escribir:

sin(pi/2)

R imprime correctamente :

[1] 1

## **1.4 Guardar entrada y salida**

Al analizar datos o programar en R, un usuario nunca se meterá en problemas serios siempre que siga dos reglas básicas:

1. Deje siempre intactos los archivos de datos originales. Cualquier versión revisada de los datos debe escribirse en un archivo *nuevo*. Si está trabajando con un conjunto de datos particularmente grande y difícil de manejar, escriba un programa corto que se adapte a lo que necesita, guarde el archivo limpiado por separado y luego escriba el código que funcione con el nuevo archivo.
2. Escriba todo el código de entrada en una secuencia de comandos que se guarda. Los usuarios generalmente deben evitar escribir código directamente en la consola. Esto incluye código para limpiar, recodificar y remodelar datos, así como para realizar análisis o desarrollar nuevos programas.

Si se siguen estas dos reglas, entonces el usuario siempre recuperará su trabajo hasta el punto de que se produzca algún error u omisión. Entonces, incluso si, en la administración de datos, se pierde o pierde información esencial, o incluso si un revisor de una revista nombra un predictor que un modelo debe agregar, el usuario siempre puede volver sobre sus pasos. Al llamar al conjunto de datos original con el programa guardado, el usuario puede realizar ajustes *menores* en el código para incorporar una nueva función de análisis o recuperar información perdida. Por el contrario, si los datos originales se sobrescriben o el código de entrada no se guarda, entonces el usuario probablemente tendrá que iniciar el proyecto completo desde el principio, lo que es una pérdida de tiempo.

Un *archivo de script* en R es simplemente texto sin formato, generalmente guardado con el sufijo .R. Para crear un nuevo archivo de script en R, simplemente elija *Archivo Nuevo documento* en el menú desplegable para abrir el documento. Alternativamente, la ventana de la consola que se muestra en la Fig. 1.2 muestra un ícono que parece una página en blanco en la parte superior de la pantalla (segundo ícono de la derecha). Al hacer clic en esto, también se creará un nuevo archivo de script R. Una vez abierto, se aplican los comandos normales *Guardar* y *Guardar como* del menú *Archivo*. Para abrir un script existente, seleccione *Archivo Abrir documento* en el menú desplegable, o haga clic en el icono en la parte superior de la pantalla que parece un página con escritura (tercer icono desde la derecha en la Fig. 1.2). Cuando se trabaja con un archivo de secuencia de comandos, cualquier código dentro del archivo se puede ejecutar en la consola simplemente resaltando el código de interés y escribiendo el atajo de teclado Ctrl + R en Windows o Cmd + Return en Mac. Además del editor de archivos de script predeterminado, también están disponibles editores de texto más sofisticados como Emacs y RWinEdt.

El producto de cualquier sesión de R se guarda en el *directorio de trabajo*. El directorio de trabajo es la ruta de archivo predeterminada para todos los archivos que el usuario desea leer o escribir. El comando getwd (que significa \*\* get w \*\* orking \*\* d \*\* irectory) imprimirá el directorio de trabajo actual de R, mientras que setwd (\*\* set w \*\* orking \*\* d \*\* irectory) le permite cambiar el directorio de trabajo como se desee. Dentro de una máquina con Windows, la sintaxis para verificar y luego configurar, el directorio de trabajo se vería así:

getwd()   
setwd("C:/temp/")

This now writes any output files, be they data sets, figures, or printed output to the folder temp in the C: drive. Observe that R expects forward slashes to designate subdirectories, which contrasts from Windows’s typical use of backslashes. Hence, specifying C:/temp/ as the working directory points to C: in normal Windows syntax. Meanwhile for Mac or Unix, setting a working directory would be similar,andthepathdirectoryisprintedexactlyastheseoperatingsystemsdesignate them with forward slashes:

setwd("/Volumes/flashdisk/temp")

Note that setwd can be called multiple times in a session, as needed. Also, specifying the full path for any file overrides the working directory.

To *save output* from your session in R, try the sink command. As a general computing term, a **sink** is an output point for a program where data or results are written out. In R, this term accordingly refers to a file that records all of our printed output. To save your session’s ouput to the file Rintro.txt within the working directory type:

sink("Rintro.txt")

Alternatively, if we wanted to override the working directory, in Windows for instance, we could have instead typed:

sink("C:/myproject/code/Rintro.txt")

Now that we have created an output file, any output that normally would print to the console will instead print to the file Rintro.txt. (For this reason, in a first runofnewcode,itisusuallyadvisable toallowoutputtoprinttothescreenandthen rerun the code later to print to a file.) The print command is useful for creating output that can be easily followed. For instance, the command:

print("The mean of variable x is...")

will print the following in the file Rintro.txt:

[1] “The mean of variable x is…”

Another useful printing command is the cat command (short for **cat**enate, to connect things together), which lets you mix objects in R with text. As a preview of simulation tools described in Chap.11, let us create a variable named x by means of simulation:

x <- rnorm(1000)

By way of explanation: this syntax draws randomly 1000 times from a standard normal distribution and assigns the values to the vector x. Observe the arrow (<-), formed with a *less than* sign and a *hyphen*, which is R’s assignment operator. Any time we assign something with the arrow (<-) the name on the left (x in this case) allows us to recall the result of the operation on the right (rnorm(1000) in this case)[[3]](#footnote-32). Now we can print the mean of these 1000 draws (which should be close to 0 in this case) to our output file as follows:

cat("The mean of variable x is...", mean(x), "\n")

With this syntax, objects from R can be embedded into the statement you print. The character puts in a carriage return. You also can print any statistical output using the either print or cat commands. Remember, your output does not go to the log file unless you use one of the print commands. Another option is to simply copy and paste results from the R console window into Word or a text editor. To turn off the sink command, simply type:

sink()

1. **Work Session Management**

A key feature of R is that it is an *object-oriented* programming language. Variables, data frames, models, and outputs are all stored in memory as *objects*, or identified (and named) locations in memory with defined features. R stores in working memory any object you create using the name you define whenever you load data into memory or estimate a model. To **l**i**s**t the objects you have created in a session use either of the following commands:

objects()   
ls()

To **r**e**m**ove all the objects in R type:

rm(list=ls(all=TRUE))

As a rule, it is a good idea to use the rm command at the start of any new program. If the previous user saved his or her workspace, then they may have used objects sharing the same name as yours, which can create confusion.

To **q**uit R either close the console window or type:

q()

At this point, R will ask if you wish to save the workspace image. Generally, it is advisable not to do this, as starting with a clean slate in each session is more likely to prevent programming errors or confusion on the versions of objects loaded in memory.

Finally, in many R sessions, we will need to load *packages*, or batches of code and data offering additional functionality not written in R’s base code. Throughout this book we will load several packages, particularly in Chap.8, where our focus will be on example packages written by prominent Political Scientists to implement cutting-edge methods. The necessary commands to load packages are install.packages, a command that automatically downloads and installs a package on a user’s copy of R, and library, a command that loads the package in a given session. Suppose we wanted to install the package MCMCpack. This package provides tools for Bayesian modeling that we will use in Chap.8. The form of the syntax for these commands is:

install.packages("MCMCpack")   
library(MCMCpack)

Package installation is case and spelling sensitive. R will likely prompt you at this point to choose one of the CRAN mirrors from which to download this package: For faster downloading, users typically choose the mirror that is most geographically proximate. The install.packages command only needs to be run once per R installation for a particular package to be available on a machine. The library command needs to be run for every session that a user wishes to use the package. Hence, in the next session that we want to use MCMCpack, we need only type: library(MCMCpack).

1. **Resources**

Given the wide array of base functions that are available in R, much less the even wider array of functionality created by R packages, a book such as this cannot possibly address everything R is capable of doing. This book should serve as a resource introducing how a researcher can use R as a basic statistics program and offer some general pointers about the usage of packages and programming features. As questions emerge about topics not covered in this space, there are several other resources that may be of use:

* Within R, the *Help* pull down menu (also available by typing help.start() in the console) offers several manuals of use, including an “Introduction to R” and “Writing R Extensions.” This also opens an HTML-based search engine of the help files.
* UCLA’s Institute for Digital Research and Education offers several nice tutorials [(http://www.ats.ucla.edu/stat/r/)](http://www.ats.ucla.edu/stat/r/). The CRAN website also includes a variety of online manuals [(http://www.cran.r-project.org/other-docs.html)](http://www.cran.r-project.org/other-docs.html).
* Some nice interactive tutorials include swirl, which is a package you install in your own copy of R (more information:[http://www.swirlstats.com/)](http://www.swirlstats.com/), and Try R, which is completed online [(http://tryr.codeschool.com/)](http://tryr.codeschool.com/).
* Within the R console, the commands ?, help(), and help.search() all serve to find documentation. For instance, ?lm would find the documenta- tion for the linear model command. Alternatively, help.search(“linear model”) would search the documentation for a phrase.

1. Practice Problems 11

* To search the internet for information,Rseek [(http://www.rseek.org/,](http://www.rseek.org/) powered by Google) is a worthwhile search engine that searches only over websites focused on R.
* Finally, Twitter users reference R through the hashtag #rstats.

At this point, users should now have R installed on their machine, hold a basic sense of how commands are entered and output is generated, and recognize where to find the vast resources available for R users. In the next six chapters, we will see how R can be used to fill the role of a statistical analysis or econometrics software program.

**1.6 Practice Problems**

Each chapter will end with a few practice problems. If you have tested all of the code from the in-chapter examples, you should be able to complete these on your own. If you have not done so already, go ahead and install R on your machine for free and try the in-chapter code. Then try the following questions.

1. Compute the following in R:
2. 7 23
3. 8

82C 1

1. cpos
2. 81
3. ln*e*4
4. Whatdoesthecommandcor do?Finddocumentationaboutitanddescribewhat the function does.
5. What does the command runif do? Find documentation about it and describe what the function does.
6. Create a vector named x that consists of 1000 draws from a standard normal distribution, using code just like you see in Sect.[1.3.](#Xd7cf74b3ffb77015970af54861e2b84db3f97b9) Create a second vector named y in the same way. Compute the correlation coefficient between the two vectors. What result do you get, and why do you get this result?
7. Get a feel for how to decide when add-on packages might be useful for you. Log in to<http://www.rseek.org>and look up what the stringr package does. What kinds of functionality does this package give you? When might you want to use it?

# 2 **Capitulo 2: Carga y manipulación de datos**

Ahora pasamos a usar **R** para realizar análisis de datos. Nuestros primeros pasos básicos son simplemente cargar datos en**R** y limpiar los datos para que se adapten a nuestros propósitos. La limpieza y recodificación de datos son a menudo una tarea tediosa de análisis de datos, pero sin embargo son esenciales porque los datos mal codificados producirán resultados erróneos cuando se estima un modelo usándolos. (En otras palabras, basura entra, basura sale). En este capítulo, cargaremos varios tipos de datos usando diferentes comandos, veremos nuestros datos para comprender sus características, practicaremos recodificar datos para limpiarlos según sea necesario, fusionar datos conjuntos y remodelar conjuntos de datos.

Nuestro ejemplo de trabajo en este capítulo será un subconjunto de Poe et al. (1999) Los datos de Political Terror Scale sobre derechos humanos, que son una actualización de los datos de Poe y Tate (1994). Mientras que sus datos completos cubren 1976-1993, nos centramos únicamente en el año 1993. Las ocho variables que contiene este conjunto de datos son:

**país**: Una variable de carácter que enumera el país por nombre. **democ**: Puntuación del país en la escala de democracia Polity III. Las puntuaciones van desde 0 (menos democrático) a 10 (más democrático). **sdnew**: La escala de terror político del Departamento de Estado de EE. UU. Las puntuaciones van desde 1 (terrorismo de estado bajo, menor número de violaciones de la integridad personal) al 5 (mayor número de violaciones de la integridad personal). **militar**: Una variable ficticia codificada 1 para un régimen militar, 0 en caso contrario. **gnpcats**: Nivel de PNB per cápita en cinco categorías: 1 = menos de $ 1000, 2 = $ 1000– $ 1999, 3 = $ 2000– $ 2999, 4 = $ 3000– $ 3999, 5 = más de $ 4000. **lpop**: Logaritmo de la población nacional. **civ\_war**: Una variable ficticia codificada con 1 si está involucrada en una guerra civil, 0 en caso contrario. **int\_war**: Una variable ficticia codificada como 1 si está involucrada en una guerra internacional, 0 en caso contrario.

## Lectura de datos

Ingresar datos **R** es bastante fácil. Hay tres formas principales de importar datos: ingresar los datos manualmente (quizás escritos en un archivo de script), leer datos de un archivo con formato de texto e importar datos de otro programa. Dado que es un poco menos común en Ciencias Políticas, el ingreso de datos manualmente se ilustra en los ejemplos del Cap.10, en el contexto de la creación de vectores, matrices y marcos de datos. En esta sección, nos enfocamos en importar datos de archivos guardados.

Primero, consideramos cómo leer en un archivo de texto delimitado con el comando \*\*read.table\*\*. \*\*R\*\* leerá en una variedad de archivos delimitados. (¿Para todas las opciones asociadas con este tipo de comando \*\*?read.table\*\* en \*\*R\*\*.) En los datos basados en texto, normalmente cada línea representa una observación única y algún delimitador designado separa cada variable de la línea. El predeterminado para \*\*read.table\*\* es un archivo delimitado por espacios en el que cualquier espacio en blanco designa variables diferentes. Dado que nuestro Poe et al. archivo de datos, llamado \*\*hmnrghts.txt\*\*, está separado por espacios, podemos leer nuestro archivo en \*\*R\*\* usando la siguiente línea de código. Este archivo de datos está disponible en el Dataverse mencionado en la página vii o en el enlace de contenido del capítulo en la página 13. Es posible que deba utilizar \*\*setwd\*\* comando introducido en el Cap. 1 apuntar \*\*R\*\* a la carpeta donde ha guardado los datos. Después de esto, ejecute el siguiente código:

hmnrghts<-read.table("hmnrghts.txt",  
 header=TRUE, na="NA")

Nota: Como se mencionó en el capítulo anterior, **R** permite al usuario dividir un solo comando en varias líneas, lo que hemos hecho aquí. A lo largo de este libro, los comandos que abarcan varias líneas se distinguirán con sangría francesa. Pasando al código en sí, observe algunas características: Una, como se señaló en el capítulo anterior, el símbolo de la flecha izquierda (**<-**) asigna nuestro archivo de entrada a un objeto. Por eso, **hmnrghts** es el nombre que asignamos a nuestro archivo de datos, pero podríamos haberlo llamado de cualquier forma. En segundo lugar, el primer argumento del comando **read.table** llama al nombre del archivo de texto **hmnrghts.txt**. Podríamos haber precedido a este argumento con la opción **file=**, y habríamos tenido que hacerlo si no hubiéramos incluido esto como el primer argumento, pero **R** reconoce que el archivo en sí es normalmente el primer argumento que toma este comando. En tercer lugar, especificamos **header=TRUE**, lo que transmite que la primera fila de nuestro archivo de texto enumera los nombres de nuestras variables. Es esencial que este argumento se identifique correctamente, de lo contrario, los nombres de variables pueden asignarse erróneamente como datos o los datos como nombres de variables[[4]](#footnote-41). Finalmente, dentro del archivo de texto, los caracteres *NA* se escriben siempre que falta una observación de una variable. La opción **na = “NA”** transmite a **R** que este es el símbolo del conjunto de datos para un valor faltante. (Otros símbolos comunes de ausencia son un punto (*.*) o el número *-9999*.)

El comando **read.table** también tiene otras opciones importantes. Si su archivo de texto usa un delimitador que no sea un espacio, entonces esto se puede transmitir a **R** utilizando la opción **sep**. Por ejemplo, incluyendo **sep = "** en el comando anterior nos permitiría leer en un archivo de texto separado por tabulaciones, mientras **sep=“,”** habría permitido un archivo separado por comas. Los comandos **read.csv** y **read.delim** son versiones alternativas de **read.table** que simplemente tienen diferentes valores predeterminados. (Particularmente, **read.csv** está orientado a leer comma-sseparado varchivos de alues y read.delim

está orientado a leer pestaña-delimitararchivos incluidos, aunque algunos otros valores predeterminados también cambian.) Otra opción importante para estos comandos es cita. Los valores predeterminados varían en estos comandos para los cuales los caracteres designan variables basadas en cadenas que usan texto alfabético como valores, como el nombre de la observación (por ejemplo, país, estado, candidato). Laread.table El comando, por defecto, usa comillas simples o dobles alrededor de la entrada en el archivo de texto. Esto sería un problema si se usaran comillas dobles para designar el texto, pero hubiera apóstrofes en el texto. Para compensar, simplemente especifique la opcióncita = “"” para permitir solo comillas dobles. (Observe la barra invertida para indicar que la comilla doble es un argumento). Alternativamente,read.csv y read.delim Ambos solo permiten comillas dobles de forma predeterminada, por lo que se especifica cita = “" ’” permitiría comillas simples o dobles, o cita = “ ’”cambiaría a comillas simples. Los autores también pueden especificar otros caracteres en esta opción, si es necesario. Una vez que descargue un archivo, tiene la opción de especificar el directorio de ruta completo en el comando para abrir el archivo. Supongamos que hubiéramos salvadohmnrghts.txt en el directorio de ruta C: / temp /, entonces podríamos cargar el archivo de la siguiente manera:

hmnrghts <- read.table (“C: /temp/hmnrghts.txt,” encabezado = VE**R**DADE**R**O, na = “NA”)

Como se mencionó cuando cargamos el archivo por primera vez, otra opción habría sido usar la setwd comando para colocar la working Directory, lo que significa que no tendríamos que enumerar la ruta completa del archivo en la llamada a read.table. (Si hacemos esto, todos los archivos de entrada y salida irán a este directorio, a menos que especifiquemos lo contrario). Finalmente, en cualquier sistema basado en GUI (por ejemplo, no terminal), podríamos escribir:

hmnrghts <-read.table (file.choose (), header = T**R**UE, na = “NA”)

La file.choose () La opción abrirá un navegador de archivos que permitirá al usuario localizar y seleccionar el archivo de datos deseado, que **R** luego se asignará al objeto nombrado en la memoria (hmnrghts en este caso). Esta opción de navegador es útil en el análisis interactivo, pero menos útil para programas automatizados. Otro formato de datos basados en texto es un archivo de ancho fijo. Los archivos de este formato no utilizan un carácter para delimitar variables dentro de una observación. En cambio, ciertas columnas de texto están constantemente dedicadas a una variable. Por lo tanto,**R** necesita saber qué columnas definen cada variable para leer en los datos. El comando paraleering un Fijo width File es read.fwf. Como una ilustración rápida de cómo cargar este tipo de datos, cargaremos un conjunto de datos diferente: votaciones nominales del 113 ° Senado de los Estados Unidos, el período que se extiende desde 2013 hasta 2015.2 Este conjunto de datos se revisará en un problema de práctica.

2Estos datos fueron recopilados por Jeff Lewis y Keith Poole. Para más información, verhttp: // www. voteview.com/senate113.htm.

dieciséis 2 Carga y manipulación de datos

en el Cap. 8. Para abrir estos datos, comience por descargar el archivosen113kh.ord del Dataverse que aparece en la página vii o del enlace del contenido del capítulo en la página 13. Luego, escriba: senate.113 <-read.fwf (“sen113kh.ord,” anchos = c (3,5,2,2,8,3,1,1,11, rep (1,657)))

El primer argumento de read.fwf es el nombre del archivo, que extraemos de una U**R**L. (La extensión del archivo es.ord, pero el formato es texto sin formato. Intente abrir el archivo en el Bloc de notas o TextEdit solo para familiarizarse con el formato.) El segundo argumento, anchos, es esencial. Para cada variable, debemos ingresar el número de caracteres asignados a esa variable. En otras palabras, la primera variable tiene tres caracteres, la segunda tiene cinco caracteres y así sucesivamente. Este procedimiento debería dejar en claro que debemos tener un libro de códigos para un archivo de ancho fijo, o ingresar los datos es un esfuerzo inútil. Observe que el último componente de laanchos el argumento es rep (1.657). Esto significa que nuestro conjunto de datos termina con 657 variables de un carácter. Estos son los 657 votos que emitió el Senado durante ese período del Congreso, con cada variable registrando si cada senador votó sí, no, presente o no votó.

Con cualquier tipo de archivo de datos, incluido el ancho fijo, si el archivo en sí no tiene los nombres de las variables, podemos agregarlos en **R**. (Nuevamente, aquí es útil un buen libro de códigos). read.table, read.csv, y read.fwf todos incluyen una opción llamada nombres de col. que permite al usuario nombrar cada variable en el conjunto de datos al leer en el archivo. Sin embargo, en el caso de las votaciones nominales del Senado, es más fácil para nosotros nombrar las variables después de la siguiente manera: colnames (senate.113) [1: 9] <- c (“congreso,” “icpsr,” “código de estado,” “cd,” “state.name,” “party,” “occupancy,” “attaining,” “name”) for (i en 1: 657) {colnames (senate.113) [i + 9] <- pegar ( “**R**C,” i, sep = "")}

En este caso, usamos el colnames comando para establecer los nombres de las variables. A la izquierda de la flecha, especificando [1: 9], indicamos que solo estamos nombrando las primeras nueve variables. A la derecha de la flecha, usamos uno de losmas fundamental comandos en **R**: la Com bine comandoC), que combina varios elementos en un vector. En este caso, nuestro vector incluye los nombres de las variables en texto. En la segunda línea, procedemos a nombrar los 657 votos nominales.**R**C1, **R**C2,. . . , **R**C657. Para ahorrar tipeo, hacemos esta asignación usando un por loop, que se describe con más detalle en el Cap. 11. Dentro depor bucle, usamos el pegar comando, que simplemente imprime nuestro texto (“**R**C”) y el número de índice I, sepclasificado por nada (de ahí las comillas vacías al final). Por supuesto, por defecto,**R** asigna nombres de variables genéricas (V1, V2, etc.), por lo que un lector que se contente con usar nombres genéricos puede omitir este paso, si lo prefiere. (Tenga en cuenta, sin embargo, que si nombramos las primeras nueve variables como lo hicimos, la primera votación nominal se llamaríaV10 sin que apliquemos un nuevo nombre).

2.1 Lectura de datos 17

2.1.1 Lectura de datos de otros programas

Volviendo a nuestro ejemplo de derechos humanos, también puede importar datos de muchos otros programas estadísticos. Una de las bibliotecas más importantes de**R** es el extranjero , lo que facilita la incorporación de datos de otros paquetes estadísticos, como SPSS, Stata y Minitab.3 Como alternativa a la versión de texto de los datos de derechos humanos, también podríamos cargar un archivo de datos con formato Stata, hmnrghts.dta. Los archivos de estadísticas generalmente tienen una extensión de archivo de dta, que es lo que el read.dta comando se refiere. (Similar,read.spss voluntad leer un SPSS-archivo formateado con el.sav extensión de archivo.) Para abrir nuestros datos en formato Stata, necesitamos descargar el archivo hmnrghts.dta desde el Dataverse vinculado en la página vii o el contenido del capítulo vinculado en la página 13. Una vez que lo guardamos en nuestro disco duro, podemos configurar nuestro directorio de trabajo, listar la ruta completa del archivo o usar el file.choose () comando para acceder a nuestros datos. Por ejemplo, si descargamos el archivo, que se llamahmnrghts.dta, en nuestro C:  temp  carpeta, podríamos abrirla escribiendo: biblioteca (extranjera) setwd (“C: / temp /”) hmnrghts.2 <- read.dta (“hmnrghts.dta”)

Cualquier dato en formato Stata que seleccione se convertirá a **R** formato. Una palabra de advertencia, por defecto si hay etiquetas de valor en datos con formato Stata,**R** importará las etiquetas como una variable con formato de cadena. Si esto es un problema para usted, intente importar los datos sin etiquetas de valor para guardar las variables usando los códigos numéricos. Vea el comienzo del Cap.7 para un ejemplo de la convert.factors = FALSE opción. (Una opción para los conjuntos de datos que no son excesivamente grandes es cargar dos copias de un conjunto de datos de Stata, una con las etiquetas como texto para que sirva como un libro de códigos y otra con códigos numéricos para el análisis). Siempre es bueno ver exactamente cómo los datos se formatean inspeccionando la hoja de cálculo después de importar con las herramientas descritas en la Sección.2.2.

2.1.2 Tramas de datos en **R**

**R** distingue entre vectores, listas, marcos de datos, y matrices. Cada uno de estos es un objeto de una clase diferente en **R**. Los vectores están indexados por longitud y las matrices están indexadas por filas y columnas. Las listas no son omnipresentes para los análisis básicos, pero son útiles para el almacenamiento complejo y, a menudo, se las considera comogenérico vectores donde cada elemento puede ser cualquier clase de objeto. (Por ejemplo, una lista podría ser un vector de

3La extranjero El paquete se usa con tanta frecuencia que ahora se descarga con cualquier **R** instalación. Sin embargo, en el improbable caso de que el paquete no se cargue con elBiblioteca comando, simplemente escriba install.packages (“extranjero”) en el símbolo del sistema para descargarlo. Alternativamente, para los usuarios que deseen importar datos deSobresalir, hay dos opciones: una es guardar el archivo de Excel en formato de valores separados por comas y luego usar el read.csv mando. El otro es instalar elXLConnect biblioteca y use la readWorksheetFromFile mando.

18 2 Carga y manipulación de datos

resultados del modelo, o una combinación de marcos de datos y mapas.) Un marco de datos es una matriz que **R** designa como un conjunto de datos. Con un marco de datos, las columnas de la matriz pueden denominarse variables. Después de leer un conjunto de datos,**R** tratará sus datos como un marco de datos, lo que significa que puede hacer referencia a cualquier variable dentro de un marco de datos agregando $ VA**R**IABLENAME al nombre del marco de datos.4 Por ejemplo, en nuestros datos de derechos humanos podemos imprimir la variable país para ver qué países están en el conjunto de datos: hmnrghts $ país Otra opción para llamar a variables, aunque un desaconsejable uno, es usar el adjuntar mando. **R** permite al usuario cargar múltiples conjuntos de datos a la vez (en contraste con algunos de los programas de análisis de datos disponibles comercialmente). Laadjuntar El comando coloca un conjunto de datos al frente y permite al usuario llamar directamente los nombres de las variables sin hacer referencia al nombre del marco de datos. Por ejemplo: adjuntar (hmnrghts) país Con este código, **R** reconocería país de forma aislada como parte del conjunto de datos adjunto e imprímalo como en el ejemplo anterior. El problema con este enfoque es que**R** puede almacenar objetos en la memoria con el mismo nombre como algunas de las variables del conjunto de datos. De hecho, al recodificar datos, el usuario debesiempre referirse al marco de datos por su nombre, de lo contrario **R** confusamente creará una copia de la variable en la memoria que es distinta de la copia en el marco de datos. Por esta razón, generalmente recomiendo no adjuntar datos. Si, por alguna circunstancia, un usuario siente que adjuntar un marco de datos es inevitable, entonces el usuario puede realizar lo que debe hacerse con los datos adjuntos y luego usar eldespegar comando lo antes posible. Este comando funciona como era de esperar, eliminando la designación de un marco de datos de trabajo y ya no permite que el usuario llame a las variables de forma aislada: separar (hmnrghts)

2.1.3 Escritura de datos

Para exportar los datos que está utilizando en **R** a un archivo de texto, use las funciones escribir.tabla o write.csv. Dentro de extranjero Biblioteca, write.dta permite al usuario escribir un archivo con formato Stata. Como ejemplo simple, podemos generar una matriz con cuatro observaciones y seis variables, contando de 1 a 24. Luego, podemos escribir esto en un archivo de valores separados por comas, un archivo de texto delimitado por espacios y un archivo Stata: x <- matriz (1:24, nrow = 4) write.csv (x, file = “sample.csv”) write.table (x, file = “sample.txt”) write.dta (as.data.frame (x), archivo = “sample.dta”)

4Más técnicamente, los marcos de datos son objetos del S3 clase. Para todosS3 objetos, los atributos del objeto (como las variables) se pueden llamar con el signo de dólar ($).

2.2 Visualización de atributos de los datos 19

Tenga en cuenta que el comando as.data.frame convierte matrices en marcos de datos, una distinción descrita en la sección anterior. El comandowrite.dta espera que el objeto sea del marco de datos clase. No es necesario realizar esta conversión si el objeto ya está formateado como datos, en lugar de una matriz. Para comprobar su esfuerzo para guardar los archivos, intente eliminarX de la memoria con el comando rm (x) y luego restaurar los datos de uno de los archivos guardados. Para mantenerse al día con el destino de los archivos de datos guardados, los archivos que escribimos se guardarán en nuestro directorio de trabajo. A obtener la working Directorio (es decir, tener **R** dinos qué es), simplemente escribe: getwd (). Para cambiar el directorio de trabajo donde se escribirán los archivos de salida, podemos colocar la working Directory, usando el mismo setwd comando que consideramos al abrir archivos anteriormente. Todos los archivos guardados posteriormente se enviarán al directorio especificado, a menos que**R** se dice explícitamente lo contrario.

2.2 Visualización de atributos de los datos

Una vez que se ingresan los datos en **R**, la primera tarea debería ser inspeccionar los datos y asegurarse de que se hayan cargado correctamente. Con un conjunto de datos relativamente pequeño, simplemente podemos imprimir todo el marco de datos en la pantalla: hmnrghts

La impresión del conjunto de datos completo, por supuesto, no se recomienda ni es útil con conjuntos de datos grandes. Otra opción es mirar los nombres de las variables y las primeras líneas de datos para ver si los datos están estructurados correctamente a través de algunas observaciones. Esto se hace con elcabeza mando: cabeza (hmnrghts)

Para obtener una lista rápida de los nombres de las variables en nuestro conjunto de datos (que también puede ser útil si se olvida la ortografía exacta o el uso de mayúsculas en los nombres de las variables) escriba: nombres (hmnrghts)

Una ruta para obtener una visión completa de los datos es utilizar la reparar mando: arreglar (hmnrghts)

Esto presenta los datos en una hoja de cálculo que permite una vista rápida de las observaciones o variables de interés, así como la oportunidad de ver que la matriz de datos se cargó correctamente. Un ejemplo de esta ventana del editor de datos quereparar abre se presenta en la Fig. 2.1. El usuario tiene la opción de editar datos dentro de la ventana de la hoja de cálculo quereparar crea, aunque a menos que los datos revisados se escriban en un nuevo archivo, no habrá un registro permanente de estos cambios.5 Además, es clave tener en cuenta que antes de continuar un **R** sesión

5La Vista el comando es similar a reparar, pero no permite la edición de observaciones. Si prefiere solo poder ver los datos sin editar valores (tal vez incluso apoyándose accidentalmente en su teclado), entoncesVista podría ser preferible.

20 2 Carga y manipulación de datos

Figura 2.1 **R** editor de datos abierto con el reparar mando

con más comandos, debe cerrar la ventana del editor de datos. La consola está congelada mientras elreparar la ventana está abierta. Hablaremos más sobre estadística descriptiva en el Cap. 4. Mientras tanto, sin embargo, Puede ser informativo ver algunos de los estadísticos descriptivos básicos (incluida la media, la mediana, el mínimo y el máximo), así como un recuento del número de observaciones faltantes para cada variable: resumen (hmnrghts)

Alternativamente, esta información se puede obtener solo para una sola variable, como la población registrada: resumen (hmnrghts $ lpop)

2.3 Declaraciones lógicas y generación de variables

A medida que pasamos a limpiar los datos que se cargan en **R**, un conjunto de herramientas esencial es el grupo de declaraciones lógicas. Declaraciones lógicas (o booleanas) en**R** son evaluados en cuanto a si son CIE**R**TO o FALSO. Mesa 2.1 resume los operadores lógicos comunes en **R**.

2.3 Declaraciones lógicas y generación de variables 21

Cuadro 2.1 Operadores lógicos en **R** Operador Medio < Menos que <= Menor o igual a > Mayor que > = Mayor o igual a Igual a ==  
! = No igual a Y Y | O

Tenga en cuenta que la declaración booleana “es igual a” se designa con dos signos iguales (==), mientras que un solo signo igual (=) sirve como operador de asignación. Para aplicar algunos de estos operadores booleanos de la tabla 2.1 en la práctica, supongamos, por ejemplo, que quisiéramos saber qué países estaban en una guerra civil y tenían un puntaje de democracia superior al promedio en 1993. Podríamos generar una nueva variable en nuestro conjunto de datos de trabajo, que llamaré dem.civaunque el usuario puede elegir el nombre). Luego, podemos ver una tabla de nuestra nueva variable y listar todos los países que se ajustan a estos criterios: hmnrghts $ dem.civ <- as.numeric (hmnrghts $ civ\_war == 1 & hmnrghts $ democ> 5.3) tabla (hmnrghts $ dem.civ) hmnrghts $ país [hmnrghts $ dem.civ == 1]

En la primera línea, hmnrghts $ dem.civ define nuestra nueva variable dentro del conjunto de datos de derechos humanos.6 A la derecha, tenemos una declaración booleana de dos partes: la primera pregunta si el país está en una guerra civil y la segunda pregunta si el puntaje de democracia del país es más alto que el promedio de 5.3. El ampersand (&) requiere que ambas declaraciones sean verdaderas simultáneamente para que toda la declaración sea verdadera. Todo esto está incrustado en elas.numeric comando, que codifica nuestra salida booleana como a numérico variable. Específicamente, todos los valores deCIE**R**TO se establecen en 1 y FALSO los valores se establecen en 0. Esta codificación suele ser más conveniente para fines de modelado. La siguiente línea nos da una tabla de las frecuencias relativas de 0 y 1. **R**esulta que solo cuatro países tenían niveles de democracia por encima del promedio y estuvieron involucrados en una guerra civil en 1993. Para ver qué países, la última línea pregunta**R** para imprimir los nombres de los países, pero los corchetes que siguen al vector indican qué observaciones imprimir: Solo las que puntúan 1 en esta nueva variable.7

6Sin embargo, tenga en cuenta que las nuevas variables que creamos, las observaciones que eliminamos o las variables que recodificamos solo cambian los datos en memoria de trabajo. Por lo tanto, nuestro archivo de datos original en disco permanece sin cambios y, por lo tanto, es seguro para la recuperación. Nuevamente, debemos usar uno de los comandos de Sect.2.1.3 si queremos guardar una segunda copia de los datos, incluidos todos nuestros cambios. 7La salida imprime los cuatro nombres de países y cuatro valores de N / A. Esto significa que en cuatro casos, una de las declaraciones de dos componentes fue CIE**R**TO pero la otra declaración no se pudo evaluar porque faltaba la variable.

Otro tipo de declaración lógica en **R** que puede ser útil es el es declaración. Estas declaraciones preguntan si una observación u objeto cumple algún criterio. Por ejemplo,is.na es un caso especial que pregunta si falta una observación o no. Alternativamente, declaraciones comoes.matriz o is.data.frame pregunte si un objeto es de cierta clase. Considere tres ejemplos: tabla (is.na (hmnrghts $ democ)) is.matrix (hmnrghts) is.data.frame (hmnrghts) La primera declaración pregunta por cada observación si falta el valor de la democracia. Lamesa El comando luego agrega esto y nos informa que faltan 31 observaciones. Las siguientes dos declaraciones preguntan si nuestro conjunto de datoshmnrghts se guarda como una matriz, luego como un marco de datos. Laes.matriz declaración devuelve FALSO, lo que indica que los comandos basados en matrices no funcionarán en nuestros datos, y el is.data.frame declaración devuelve CIE**R**TO, lo que indica que se almacena como un marco de datos. Con un sentido de declaraciones lógicas en**R**, ahora podemos aplicarlos a la tarea de limpiar los datos.

2.4 Limpieza de datos

Una de las primeras tareas de la limpieza de datos es decidir cómo lidiar con datos perdidos. **R** designa los valores perdidos con N / A. Traduce los valores perdidos de otros paquetes de estadísticas al N / A formato faltante. Independientemente de cómo un investigador maneje los datos faltantes, es importante tener en cuenta la proporción relativa de valores no observados en los datos y qué información se puede perder. Una opción (algo burda) para lidiar con la falta sería podar el conjunto de datos mediante la eliminación por lista, o eliminar todas las observaciones para las que no se registra una sola variable. Para crear un nuevo conjunto de datos que pode de esta manera, escriba: hmnrghts.trim <- na.omit (hmnrghts)

Esto disminuye el número de observaciones de 158 a 127, por lo que se ha perdido una cantidad tangible de información. La mayoría de los comandos de modelado en **R** Brinde a los usuarios la opción de estimar el modelo solo sobre observaciones completas, implementando la eliminación por lista sobre la marcha. Como advertencia, la eliminación por lista es en realidad el valor predeterminado en los comandos básicos para los modelos lineales y lineales generalizados, por lo que la pérdida de datos puede pasar desapercibida si el usuario no tiene cuidado. Se insta a los usuarios con una sólida formación en modelos basados en regresiones a que consideren métodos alternativos para tratar los datos faltantes que sean superiores a la eliminación por listas. En particular, elratones y Amelia las bibliotecas implementan la útil técnica de la imputación múltiple (para obtener más información, consulte King et al. 2001; Little y **R**ubin1987; Frotar1987). Si, por alguna razón, el usuario necesita volver a designar los valores faltantes como si tuvieran algún valor numérico, el is.na El comando puede ser útil. Por ejemplo, si fuera beneficioso enumerar los valores faltantes como9999, entonces estos podrían codificarse como: hmnrghts $ democ [is.na (hmnrghts $ democ)] <- -9999

En otras palabras, todos los valores de la democracia para los que falta el valor tomarán el valor de 9999. Ten cuidado, aunque, como **R** y todos sus comandos de modelado ahora considerarán el valor anteriormente perdido como una observación válida e insertarán el valor engañoso de 9999 en cualquier análisis. Este tipo de acción solo debe tomarse si se requiere para la gestión de datos, un tipo especial de modelo en el que los valores extraños se pueden tachar, o el raro caso en el que una observación faltante en realidad puede adquirir un valor significativo (por ejemplo, un presupuesto conjunto de datos donde los elementos faltantes representan un gasto de $ 0).

2.4.1 Subconjuntos de datos

En muchos casos, es conveniente crear subconjuntos de nuestros datos. Esto puede significar que solo queremos observaciones de un cierto tipo, o puede significar que deseamos reducir el número de variables en el marco de datos, quizás porque los datos incluyen muchas variables que no son de interés. Si, en nuestros datos de derechos humanos, solo quisiéramos centrarnos en los países que tenían una puntuación de democracia de 6 a 10, podríamos llamar a este subconjuntodem.rights y créelo de la siguiente manera:

dem.rights <- subconjunto (hmnrghts, subconjunto = democ> 5)

Esto crea un subconjunto de 73 observaciones de nuestros datos originales. Tenga en cuenta que las observaciones con undesaparecido (N / A) valor de democ no se incluirá en el subconjunto. Las observaciones faltantes también se excluirían si hiciéramos una declaración mayor o igual a.8

Como ejemplo de selección de variables, si quisiéramos centrarnos solo en la democracia y la riqueza, podríamos mantener solo estas dos variables y un índice para todas las observaciones: dem.wealth <-subset (hmnrghts, select = c (país, democ, gnpcats))

Un medio alternativo de seleccionar las variables que deseamos conservar es utilizar un signo menos después de la Seleccione y enumere solo las columnas que deseamos eliminar. Por ejemplo, si quisiéramos todas las variables excepto los dos indicadores de si un país está en guerra, podríamos escribir: no.war <- subconjunto (hmnrghts, select = -c (civ\_war, int\_war))

Además, los usuarios tienen la opción de llamar tanto al subconjunto y Seleccione opciones si desean elegir un subconjunto de variables sobre un conjunto específico de observaciones.

8Esto contrasta con programas como Stata, que tratan los valores perdidos como infinitos positivos. En Stata, si se incluyen las observaciones faltantes depende del tipo de declaración booleana que se haga. **R** es más consistente en que los casos que faltan siempre se excluyen.

2.4.2 **R**ecodificar variables

Un aspecto final de la limpieza de datos que surge a menudo es la necesidad de recodificar las variables. Esto puede surgir porque la forma funcional de un modelo requiere una transformación de una variable, como un logaritmo o un cuadrado. Alternativamente, algunos de los valores de los datos pueden ser engañosos y, por lo tanto, deben ser recodificados como faltantes u otro valor. Otra posibilidad más es que las variables de dos conjuntos de datos deban codificarse en la misma escala: por ejemplo, si un analista ajusta un modelo con datos de encuestas y luego hace pronósticos utilizando datos del censo, entonces las variables de la encuesta y del censo deben codificarse del mismo camino. Para las transformaciones matemáticas de variables, la sintaxis es sencilla y sigue la forma del ejemplo siguiente. Supongamos que queremos la población real de cada país en lugar de su logaritmo: hmnrghts $ pop <- exp (hmnrghts $ lpop) Simplemente, estamos aplicando la función exponencial (Exp) a un valor registrado para recuperar el valor original. Sin embargo, cualquier tipo de operador matemático podría sustituirse porExp. Una variable se puede elevar al cuadrado (ˆ2̂), registradoIniciar sesión()), tomar la raíz cuadradasqrt ()), etc. La suma, resta, multiplicación y división también son válidas, ya sea con un escalar de interés o con otra variable. Supongamos que quisiéramos crear una variable ordinal codificada con 2 si un país estaba en una guerra civil y en una guerra internacional, 1 si estuvo involucrado en cualquiera de ellas y 0 si no estuvo involucrado en ninguna guerra. Podríamos crear esto agregando las variables de guerra civil y guerra internacional: hmnrghts $ war.ord <-hmnrghts $ civ\_war + hmnrghts $ int\_war

Sin embargo, una tabla rápida de nuestra nueva variable revela que ninguna nación tuvo ambos tipos de conflicto en 1993. Otro problema común que se debe abordar es cuando los datos se presentan en un formato no deseado. Nuestra variablegnpcats en realidad está codificado como una variable de texto. Sin embargo, es posible que deseemos recodificar esto como una variable ordinal numérica. Hay dos formas de lograr esto. El primero, aunque toma varias líneas de código, se puede completar rápidamente con una buena cantidad de copiar y pegar: hmnrghts $ gnp.ord <- NA hmnrghts $ gnp.ord [hmnrghts $ gnpcats == “<1000”] <- 1 hmnrghts $ gnp.ord [hmnrghts $ gnpcats == “1000-1999”] <- 2 hmnrghts $ gnp.ord [hmnrghts $ gnpcats == “2000-2999”] <- 3 hmnrghts $ gnp.ord [hmnrghts $ gnpcats == “3000-3999”] <- 4 hmnrghts $ gnp.ord [hmnrghts $ gnpcats == “> 4000”] <- 5 Aquí, se creó una variable en blanco, y luego los valores de la nueva variable se completaron dependiendo de los valores de la antigua usando declaraciones booleanas. Una segunda opción para recodificar los datos GNP se puede lograr a través de John Fox’s Cacompañar a arespondido rla egresióncarro) Biblioteca. Como biblioteca escrita por el usuario, debemos descargarla e instalarla antes del primer uso. La instalación de una biblioteca es sencilla. Primero, escriba: install.packages (“coche”)

Una vez que la biblioteca está instalada (nuevamente, un paso que no necesita repetirse a menos que **R** se reinstala), las siguientes líneas generarán nuestra medida del PNB per cápita recodificada: biblioteca (coche) hmnrghts $ gnp.ord.2 <-recode (hmnrghts $ gnpcats, ‘“<1000” = 1; “1000-1999” = 2; “2000-2999” = 3; “3000-3999” = 4; “> 4000” = 5’) Tenga cuidado de que el recodificar el mando es delicado. Entre los apóstrofos, todas las reasignaciones de valores antiguos a nuevos se definen separadas por punto y coma. Un solo espacio entre los apóstrofos generará un error. A pesar de esto, recodificar puede ahorrar a los usuarios un tiempo considerable en la limpieza de datos. La sintaxis básica de recodificar, por supuesto, podría usarse para crear variables ficticias, variables ordinales o una variedad de otras variables recodificadas. Así que ahora dos métodos han creado una nueva variable, cada una codificada del 1 al 5, donde 5 representa el PNB per cápita más alto. Otro tipo estándar de recodificación que podríamos querer hacer es crear una variable ficticia que se codifique como 1 si la observación cumple ciertas condiciones y 0 en caso contrario. Por ejemplo, supongamos que en lugar de tener categorías de PNB, solo queremos comparar la categoría más alta de PNB con todas las demás: hmnrghts $ gnp.dummy <-as.numeric (hmnrghts $ gnpcats == “> 4000”)

Al igual que con nuestro ejemplo anterior de encontrar democracias involucradas en una guerra civil, aquí usamos una declaración lógica y la modificamos con el as.numeric declaración, que convierte cada CIE**R**TO en un 1 y cada uno FALSO en un 0. Variables categóricas en **R** se le puede dar una designación especial como factores. Si designa una variable categórica como factor, **R** lo tratará como tal en la operación estadística y creará variables ficticias para cada nivel cuando se utilice en una regresión. Si importa una variable sin codificación numérica,**R** llamará automáticamente a la variable a personaje vector, y convierta el vector de caracteres en un factor en la mayoría de los comandos de análisis. Sin embargo, si lo preferimos, podemos designar que una variable es un factor de antemano y abrir una variedad de comandos útiles. Por ejemplo, podemos designarpaís como factor: hmnrghts $ país <- as.factor (hmnrghts $ país) niveles (hmnrghts $ país) Darse cuenta de **R** permite al usuario poner la misma cantidad (en este caso, la variable país) a ambos lados de un operador de asignación. Esta asignación recursiva toma los valores antiguos de una cantidad, hace que el lado derecho cambie y luego reemplaza los nuevos valores en el mismo lugar en la memoria. Laniveles El comando nos revela los diferentes valores registrados del factor. Para cambiar qué nivel es el primer nivel (por ejemplo, para cambiar qué categoría **R** utilizará como categoría de referencia en una regresión) utilice el relevel mando. El siguiente código establece “estados unidos” como la categoría de referencia parapaís: hmnrghts $ país <-relevel (hmnrghts $ país, “estados unidos”) niveles (hmnrghts $ país)

Ahora, cuando vemos los niveles del factor, “estados unidos” aparece como el primer nivel, y el primer nivel es siempre nuestro grupo de referencia.

2.5 Fusionar y remodelar datos

Dos tareas finales que son comunes a la gestión de datos son fusionar conjuntos de datos y remodelar los datos del panel. Al considerar ejemplos de estas dos tareas, consideremos una actualización de Poe et al. (1999) datos: Específicamente, Gibney et al. (2013) han continuado codificando datos para la Escala de Terror Político. Usaremos las oleadas de 1994 y 1995 de los datos actualizados. Las variables en estas ondas son: País: Una variable de carácter que enumera el país por nombre. COWAlpha: Abreviatura de país de tres caracteres de Correlates of War conjunto de datos. VACA: Variable numérica de identificación de países del conjunto de datos Correlates of War. Banco Mundial: Abreviatura de país de tres caracteres utilizada por el Banco Mundial. Amnistía 1994 / Amnistía 1995: La escala de terror político de Amnistía Internacional. Las puntuaciones van desde 1 (terrorismo de estado bajo, menor número de violaciones de la integridad personal) a 5 (mayor número de violaciones de la integridad personal). Departamento de Estado 1994 / Departamento de Estado 1995: La escala de políticas del Departamento de Estado de EE. terror. Las puntuaciones van desde 1 (terrorismo de estado bajo, menor número de violaciones de la integridad personal) a 5 (mayor número de violaciones de la integridad personal). Para las dos últimas variables, el nombre de la variable depende de qué ola de datos se esté estudiando, y el sufijo indica el año. Tenga en cuenta que estos datos tienen cuatro variables de identificación: está diseñado explícitamente para facilitar el uso de estos datos por parte de los investigadores. Cada índice facilita que un investigador vincule estas medidas de terror político con la información proporcionada por el Banco Mundial o el conjunto de datos de Correlates of War. Esto debería mostrar cuán omnipresente es el acto de fusionar datos para la investigación de Ciencias Políticas. Con este fin, practiquemos fusionando datos. En general, la combinación de datos es útil cuando el analista tiene dos marcos de datos separados que contienen información sobre las mismas observaciones. Por ejemplo, si un politólogo tuviera un marco de datos con datos económicos por país y un segundo marco de datos que contenga los resultados electorales por país, el académico podría querer fusionar los dos conjuntos de datos para vincular los factores económicos y políticos dentro de cada país. En nuestro caso, supongamos que simplemente quisiéramos vincular el puntaje de terror político de cada país de 1994 a su puntaje de terror político de 1995. Primero, descargue los conjuntos de datos pts1994.csv y pts1995.csv del Dataverse en la página vii o el enlace del contenido del capítulo en la página 13. Como antes, es posible que deba usar setwd apuntar **R** a la carpeta donde ha guardado los datos. Después de esto, ejecute el siguiente código para cargar los datos relevantes: hmnrghts.94 <-read.csv (“pts1994.csv”) hmnrghts.95 <-read.csv (“pts1995.csv”) Estos datos están separados por comas, por lo que read.csv es el mejor comando en este caso. Si quisiéramos echar un vistazo a las primeras observaciones de nuestra onda de 1994, podríamos escribir cabeza (hmnrghts.94). Esto imprimirá lo siguiente:

País COWAlpha COW WorldBank 1 Estados Unidos EE.UU 2 EE.UU 2 Canadá LATA 20 LATA 3 Bahamas BHM 31 BHS 4 Cuba  
CACHO**RR**O 40  
CACHO**RR**O 5 Haití HAI 41 HTI 6 **R**epública Dominicana DOM 42 DOM Amnistía 1994 Departamento de Estado 1994 1 1 N / A 2 1 1 3 1 2 4 3 3 5 5 4 6 2 2 Similar, cabeza (hmnrghts.95) imprimirá las primeras observaciones de nuestro 1995 onda: País COWAlpha COW WorldBank 1 Estados Unidos EE.UU 2 EE.UU 2 Canadá LATA 20 LATA 3 Bahamas BHM 31 BHS 4 Cuba  
CACHO**RR**O 40  
CACHO**RR**O 5 Haití HAI 41 HTI 6 **R**epública Dominicana DOM 42 DOM Amnistía 1995 Departamento de Estado 1995 1 1 N / A 2 N / A 1 3 1 1 4 4 3 5 2 3 6 2 2 Como podemos ver en nuestra mirada a la parte superior de cada marco de datos, los datos están ordenados de manera similar en cada caso, y nuestras cuatro variables de índice tienen el mismo nombre en cada conjunto de datos respectivo. Solo necesitamos un índice para fusionar nuestros datos y los otros tres son redundantes. Por esta razón, podemos eliminar tres de las variables de índice de los datos de 1995: hmnrghts.95 <-subconjunto (hmnrghts.95, select = c (COW, Amnistía 1995, Departamento de Estado 1995)) Optamos por eliminar los tres índices de texto para fusionarlos en un índice numérico. Esta elección es arbitraria, sin embargo, porque**R** tampoco tiene problemas para fusionar una variable de carácter. (Intente replicar este ejercicio fusionandoCOWAlpha, por ejemplo.)

Para combinar nuestros datos de 1994 y 1995, pasamos ahora a la unir mando.9 Escribimos: hmnrghts.wide <-merge (x = hmnrghts.94, y = hmnrghts.95, by = c (“VACA”))

Dentro de este comando, la opción X se refiere a un conjunto de datos, mientras que y es el otro. Al lado depor opción, nombramos una variable de identificación que identifica de forma única cada observación. Lapor El comando en realidad permite a los usuarios nombrar múltiples variables si se necesitan varias para identificar de forma única cada observación: por ejemplo, si un investigador estaba fusionando datos donde la unidad de análisis era un país-año, una variable de país y una variable de año podrían ser esenciales para cada fila de forma única. En tal caso, la sintaxis podría leer,by = c (“VACA,” “año”). Como otra opción más, si los dos conjuntos de datos tuvieran el mismo índice, pero las variables se nombraron de manera diferente, **R** permite una sintaxis como, by.x = c (“VACA”), by.y = c (“cowCode”), lo que transmite que las variables de índice con nombres diferentes son el nombre. Una vez que hayamos combinado nuestros datos, podemos obtener una vista previa del producto terminado escribiendo cabeza (hmnrghts.wide). Esto imprime: VACA País COWAlpha WorldBank Amnistía 1994 1 2 Estados Unidos EE.UU EE.UU 1 2 20 Canadá LATA LATA 1 3 31 Bahamas BHM BHS 1 4 40 Cuba  
CACHO**RR**O  
CACHO**RR**O 3 5 41 Haití HAI HTI 5 6 42 **R**epública Dominicana DOM DOM 2 Departamento de Estado 1994 Amnistía 1995 Departamento de Estado 1995 1 N / A 1 N / A 2 1 N / A 1 3 2 1 1 4 3 4 3 5 4 2 3 6 2 2 2 Como podemos ver, las puntuaciones de 1994 y 1995 para Amnistía y Departamento de Estado se registran en un lugar para cada país. Por lo tanto, nuestra fusión fue exitosa. Por defecto,**R** excluye cualquier observación de cualquiera de los conjuntos de datos que no tenga un vinculado observación (por ejemplo, valor equivalente) del otro conjunto de datos. Entonces, si usa los valores predeterminados y el nuevo conjunto de datos incluye el mismo número de filas que los dos conjuntos de datos anteriores, todas las observaciones se vincularon e incluyeron. Por ejemplo, podríamos escribir: tenue (hmnrghts.94); tenue (hmnrghts.95); tenue (hmnrghts.wide) Esto nos diría rápidamente que tenemos 179 observaciones en ambas entradas, así como en el conjunto de datos de salida, lo que muestra que no perdimos ninguna observación. Otras opciones dentro unir están all.x, all.y, y todas, que le permiten especificar si se debe forzar

9junto al unir, la dplyr El paquete ofrece varios comandos de unión de datos que también puede resultarle útil, según sus necesidades.

la inclusión de todas las observaciones del conjunto de datos X, el conjunto de datos y, y de cualquier conjunto de datos, respectivamente. En este caso,**R** codificaría N / A valores para observaciones que no tenían un caso vinculado en el otro conjunto de datos. Como punto final de la gestión de datos, a veces necesitamos remodelar nuestros datos. En el caso de nuestro conjunto de datos fusionado,hmnrghts.wide, Hemos creado un conjunto de datos de panel (por ejemplo, un conjunto de datos que consta de observaciones repetidas de los mismos individuos) que se encuentra en formato amplio. Formato amplio significa que cada fila de nuestros datos define un individuo de estudio (un país) mientras que nuestras observaciones repetidas se almacenan en variables separadas (p. Ej., Amnistía 1994 y Amnistía 1995 récord de puntuaciones de Amnistía Internacional durante dos años distintos). En la mayoría de los modelos de datos de panel, necesitamos que nuestros datos estén enformato largo, o formato apilado. El formato largo significa que necesitamos dos variables de índice para identificar cada fila, una para el individuo (por ejemplo, país) y otra para el momento de la observación (por ejemplo, año). Mientras tanto, cada variable (p. Ej.,Amnistía) solo usará una columna. **R** nos permite remodelar nuestros datos de ancho a largo, o de largo a ancho. Por lo tanto, sea cual sea el formato de nuestros datos, podemos adaptarlos a nuestras necesidades. Para remodelar nuestros datos de terror político de formato amplio a largo, utilizamos el remodelar mando: hmnrghts.long <-reshape (hmnrghts.wide, varying = c (“Amnistía.1994,” “StateDept.1994,” “Amnistía.1995,” “StateDept.1995”), timevar = “año,” idvar = “VACA,” dirección = “largo,” sep = “.”)

Dentro del comando, el primer argumento es el nombre del marco de datos que deseamos remodelar. Lavariar término enumera todas las variables que representan observaciones repetidas a lo largo del tiempo. Consejo: Asegúrese de que las observaciones repetidas de la misma variable tengan el mismo prefijo nombre (p. ej., Amnistía o Departamento de Estado) y luego el sufijo (p.ej, 1994 o 1995) informa constantemente el tiempo. Latimevar término nos permite especificar el nombre de nuestro nuevo índice de tiempo, al que llamamos año. La idvar término enumera la variable que identifica de forma única a los individuos (países, en nuestro caso). Con dirección especificamos que queremos convertir nuestros datos en largo formato. Por último, elsep ofertas de comando **R** una pista de qué carácter separa nuestros prefijos y sufijos en las variables de observación repetidas: Dado que un punto (.) separa estos términos en cada uno de nuestros Amnistía y Departamento de Estado variables, lo denotamos aquí. Una vista previa del resultado puede ser visto escribiendo cabeza. Esto

huellas dactilares:

VACA

País COWAlpha WorldBank año

2.1994 2 Estados Unidos EE.UU Estados Unidos 1994 20.1994 20 Canadá LATA CAN 1994 31.1994 31 Bahamas BHM BHS 1994 40.1994 40 Cuba  
CACHO**RR**O CUB 1994 41.1994 41 Haití HAI HTI 1994 42.1994 42 **R**epública Dominicana DOM DOM 1994 Departamento de Estado de Amnistía  
2.1994 1 NA  
20.1994 1 1

31.1994 1 2 40.1994 3 3 41.1994 5 4 42.1994 2 2 Note que nosotros ahora solo tengo una variable para Amnistía y uno para StateDept. Ahora tenemos una nueva variable llamada año, así que entre VACA y año, cada fila identifica unívocamente cada país-año. Dado que los datos están ordenados de forma natural, la parte superior de nuestros datos solo muestra observaciones de 1994. Mecanografía cabeza (hmnrghts.long [hmnrghts.long $ año == 1995,]) muestra Veamos las primeras observaciones de 1995: VACA País COWAlpha WorldBank año 2.1995 2 Estados Unidos EE.UU Estados Unidos 1995 20.1995 20 Canadá LATA CAN 1995 31.1995 31 Bahamas BHM BHS 1995 40.1995 40 Cuba  
CACHO**RR**O CUB 1995 41.1995 41 Haití HAI HTI 1995 42.1995 42 **R**epública Dominicana DOM DOM 1995 Departamento de Estado de Amnistía 2.1995 1 N / A 20.1995 N / A 1 31.1995 1 1 40.1995 4 3 41.1995 2 3 42.1995 2 2 Como podemos ver, toda la información se conserva, ahora en formato largo (o apilado). Como ilustración final, suponga que hemos comenzado con un conjunto de datos que estaba en formato largo y queríamos uno en formato ancho. Para probar esto, reformaremoshmnrghts.long e intentar recrear nuestro amplio conjunto de datos original. Para hacer esto, escribimos: hmnrghts.wide.2 <-reshape (hmnrghts.long, v.names = c (“Amnistía,” “Departamento de Estado”), timevar = “año,” idvar = “VACA,” dirección = “ancho,” sep = “.”)

Algunas opciones ahora han cambiado: ahora usamos el v.nombres comando para indicar las variables que incluyen observaciones repetidas. Latimevar El parámetro ahora debe ser una variable dentro del conjunto de datos, al igual que idvar es decir, para separar a los individuos de puntos de tiempo repetidos. Nuestradirección el término es ahora amplio porque queremos convertir estos datos en formato ancho. Por último, elsep comando especifica el carácter que **R** utilizará para separar los prefijos de los sufijos en la forma final. Escribiendo cabeza (hmnrghts.wide.2) en la consola, ahora verá que este nuevo conjunto de datos recrea el conjunto de datos ancho original. Este capítulo ha cubierto la variedad de medios de importar y exportar datos en **R**. También ha analizado cuestiones de gestión de datos como valores perdidos, subconjuntos, recodificación de datos, fusión de datos y remodelación de datos. Con la capacidad de limpiar y administrar datos, ahora estamos listos para comenzar a analizar nuestros datos. A continuación, procedemos a la visualización de datos.

2.6 Problemas de práctica 31

2.6 Problemas de práctica

Como conjunto de datos de práctica, descargaremos y abriremos un subconjunto del Estudio Electoral Nacional Estadounidense de 2004 utilizado por Hanmer y Kalkan (2013). Este conjunto de datos se llama hanmerKalkanANES.dta, y está disponible en el Dataverse al que se hace referencia en la página vii o en el enlace de contenido del capítulo en la página 13. Estos datos están en formato Stata, así que asegúrese de cargar la biblioteca correcta y use el comando correcto al abrir. (Insinuación: Cuando utilice el comando adecuado, asegúrese de especificar el convert.factors = F opción dentro de él para obtener un resultado más fácil de leer.) Todas las variables en este conjunto de datos se relacionan con las elecciones presidenciales de EE. UU. de 2004, y son: un número de identificación del encuestado ( Identificación del caso), evaluaciones económicas retrospectivasretecon), evaluación de George El manejo de W. Bush de la guerra en Irak (bushiraq), un indicador de si el encuestado votó por Bush (voto previo), partidismo en una escala de siete puntos (partyid), ideología en una escala de siete puntos (ideol7b), un indicador de si el encuestado es blanco ( blanco), un indicador de si el encuestado es mujer (mujer), edad del encuestadoedad), nivel de educación en una escala de siete puntos (educ1\_7), e ingresos en una escala de 23 puntos ( ingreso). (La variable exptrnout2 puede ser ignorado.) 1. Una vez que haya cargado los datos, haga lo siguiente para verificar su trabajo:

1. Si preguntas **R** para devolver los nombres de las variables, ¿qué dice la lista? ¿Es correcto?
2. Usando el cabeza comando, ¿cómo se ven las primeras líneas?
3. Si usa el reparar comando, ¿los datos parecen una hoja de cálculo adecuada?
4. Utilice el resumen comando en todo el conjunto de datos. ¿Qué puedes aprender de inmediato? ¿Cuántas observaciones faltantes tienes?
5. Intente crear subconjuntos de datos de varias formas:
6. Cree una copia del conjunto de datos que elimine todas las observaciones faltantes con eliminación por lista. ¿Cuántas observaciones quedan en esta versión?
7. Cree una segunda copia que solo incluya el número de identificación del encuestado, evaluaciones económicas retrospectivas y evaluación del manejo de Bush en Irak.
8. Cree algunas variables nuevas:
9. La escala de partidismo de siete puntos (partyid) se codifica de la siguiente manera: 0 = Demócrata fuerte, 1 = Demócrata débil, 2 = Demócrata inclinado independiente, 3 = Independiente no inclinado, 4 = **R**epublicano inclinado independiente, 5 = **R**epublicano débil y 6 = **R**epublicano fuerte. Cree dos nuevas variables de indicador. El primero debe codificarse 1 si la persona se identifica como demócrata de alguna manera (incluidos los independientes que se inclinan por los demócratas) y 0 en caso contrario. La segunda variable nueva debe codificarse 1 si la persona se identifica como republicana de alguna manera (incluidos los independientes que se inclinan por la republicana) y 0 en caso contrario. Para cada una de estas dos nuevas variables, ¿quéresumen comando volver por ellos?
10. Cree una nueva variable que sea el valor al cuadrado de la edad del encuestado en años. Lo que hace elresumen retorno de comando para esta nueva variable?
11. Cree una nueva versión de la variable de ingreso que tenga solo cuatro categorías. La primera categoría debe incluir todos los valores deingreso que el rango de 1 a 12,

el segundo del 13 al 17, el tercero del 18 al 20 y el último del 21 al 23. Utilizar elmesa comando para ver la frecuencia de cada categoría. (d) Bonificación: utilice el mesa comando para comparar la versión de ingresos de 23 categorías con la versión de ingresos de cuatro categorías. ¿Codificó la nueva versión correctamente?

# 3 Visualización de datos

Palabras clave: - Marco de datos - Noticias televisivas - Paquete de celosía - Finanzas de salud - Función de gráfico

La presentación visual de datos y los resultados de los modelos se ha convertido en una pieza central del análisis político moderno. Muchas de las principales revistas de Ciencias Políticas, incluida la *Revista Estadounidense de Ciencias Políticas*, ahora solicitan gráficas en lugar de tablas siempre que ambas puedan transmitir la misma información. De hecho, Kastellec y Leoni (2007) argumentan que las gráficas transmiten resultados empíricos mejor que las tablas. Cleveland (1993) y Tufte (2001) escribió dos de los volúmenes principales que describen los elementos de una buena visualización cuantitativa, y Yau (2011) ha producido una versión más reciente de la representación gráfica. Básicamente, estos trabajos sirven como manuales de estilo para gráficos[[5]](#footnote-44). Más allá de las sugerencias que estos académicos ofrecen por el bien de los lectores, ver los propios datos visualmente transmite información sustancial sobre las características univariadas, bivariadas y multivariadas de los datos: ¿Una variable parece sesgada? ¿Parecen correlacionarse sustancialmente dos variables? ¿Cuál es la relación funcional adecuada entre variables? ¿Cómo cambia una variable en el espacio o en el tiempo? Responder estas preguntas por uno mismo como analista y por el lector en general puede elevar la calidad del análisis presentado a la disciplina.

En el límite de este movimiento gráfico en el análisis cuantitativo, **R** ofrece visualización de modelos y datos de última generación. Muchos de los programas estadísticos comerciales han intentado durante años ponerse al día con las capacidades gráficas de **R** . Este capítulo muestra estas capacidades, pasando primero a la función plot que está disponible automáticamente como parte del paquete base. En segundo lugar, analizamos algunos de los otros comandos de gráficos que se ofrecen en la biblioteca base. Finalmente, pasamos a la biblioteca lattice (o celosía), que permite al usuario crear Trellis Graphics, un marco de visualización desarrollado por Becker, Cleveland y otros para poner las sugerencias de Cleveland (1993) en la práctica. Aunque el espacio no lo permite aquí, también se anima a los usuarios a buscar los paquetes ggplot2, que ofrecen opciones de gráficos adicionales. Chang (2013), en particular, ofrece varios ejemplos de gráficos con ggplot2.

En este capítulo, trabajamos con dos conjuntos de datos de ejemplo. El primero es sobre el cabildeo en salud en los 50 estados estadounidenses, con un enfoque específico en la proporción de empresas de la industria financiera de la salud que están registradas para cabildear (Lowery et al. 2008). Una variable de predicción clave es el número total de firmas financieras de salud abiertas al público, que incluye organizaciones que brindan planes de salud, servicios comerciales, coaliciones de empleadores de salud y seguros. El conjunto de datos también incluye la tasa de participación de los grupos de presión por estado, o el número de grupos de presión como una proporción del número de empresas, no solo en el financiamiento de la salud sino para todas las empresas relacionadas con la salud y en otras seis subáreas. Estos son datos transversales del año 1997. La lista completa de variables es la siguiente:

* **stno**: Índice numérico de 1 a 50 que ordena los estados alfabéticamente.
* **raneyfolded97**: Índice de Ranney plegado de la competencia estatal bipartita en 1997[[6]](#footnote-45).
* **healthagenda97**: Número de proyectos de ley relacionados con la salud considerados por la legislatura estatal en 1997.
* **supplybusiness**: Número de establecimientos de financiación sanitaria.
* **businesssupplysq**: Número de establecimientos de financiación sanitaria al cuadrado.
* **partratebusness**: Tasa de participación en el lobby para el financiamiento de la salud: número de registros como porcentaje del número de establecimientos.
* **predecirbuspartrate**: Predicción de la tasa de participación en el financiamiento de la salud como función cuadrática del número de establecimientos de financiamiento de la salud. (Sin variables de control en la predicción).
* **partratetotalhealth**: Tasa de participación en el lobby para toda la atención médica (incluidas siete subáreas).
* **partratedpc**: Tasa de participación en el lobby para la atención directa al paciente.
* **partratepharmprod**: Tasa de participación en el lobby de medicamentos y productos sanitarios.
* **partrateprofessionals**: Tasa de participación de los profesionales de la salud en el lobby.
* **partrateadvo**: Tasa de participación del lobby para la promoción de la salud.
* **partrategov**: Tasa de participación en el lobby del gobierno local.
* **rnmedschoolpartrate**: Tasa de participación de los grupos de presión para la educación sanitaria.

En segundo lugar, analizamos las de Peake y Eshbaugh-Soha (2008) datos sobre el número de noticias de televisión relacionadas con la política energética en un mes determinado. En este marco de datos, las variables son: - **Date**: Vector de caracteres del mes y año observado. - **Energy**: Número de historias relacionadas con la energía transmitidas en los noticieros de televisión nocturnos por mes. - **Unemploy**: La tasa de desempleo por mes. - **Approval**: Aprobación presidencial por mes. - **oilc**: Precio del petróleo por barril. - **freeze1**: Variable indicadora codificada con 1 durante los meses de agosto a noviembre de 1971, cuando se impusieron congelaciones de precios y salarios. Codificado 0 en caso contrario. - **freeze2**: Variable indicadora codificada con 1 durante los meses de junio a julio de 1973, cuando se impusieron los precios, salarios y congelaciones de precios. Codificado 0 en caso contrario. - **embargo**: Una variable indicadora codificada con 1 durante los meses de octubre de 1973 a marzo de 1974, durante el embargo petrolero árabe. Codificado 0 en caso contrario. - **rehenes**: Una variable indicadora codificada con 1 durante los meses de noviembre de 1979 a enero de 1981, durante la crisis de los rehenes en Irán. Codificado 0 en caso contrario. - **Presidential speeches**: Los indicadores adicionales se codifican como 1 durante el mes en que un presidente pronunció un discurso importante sobre política energética y 0 en caso contrario. Los indicadores de los discursos respectivos se denominan: **rmn1173**, **rmn1173a**, **grf0175**, **grf575**, **grf575a**, **jec477**, **jec1177**, **jec479**, **grf0175s**, **jec479s** y **jec477s**.

## Gráficos univariados en el paquete base

Como un primer vistazo a nuestros datos, mostrar una sola variable gráficamente puede transmitir un sentido de la distribución de los datos, incluyendo su modo, dispersión, sesgo y curtosis. La biblioteca de lattice en realidad ofrece algunos comandos más para la visualización univariante que la base, pero comenzamos con los principales comandos univariados incorporados. La mayoría de los comandos de gráficos en el paquete base llaman a la función plot, pero hist y boxplot son excepciones notables.

El comando hist es útil para simplemente tener una idea de la frecuencia relativa de varios valores comunes. Comenzamos cargando nuestros datos sobre la cobertura de noticias televisivas sobre política energética. Luego creamos un **hist**ograma de esta serie temporal de conteos mensuales de historias con el comando hist. Primero, descargue los datos de Peake y Eshbaugh-Soha sobre cobertura de pólizas de energía, el archivo llamado PESenergy.csv. El archivo está disponible en el Dataverse nombrado en la página vii o en el enlace de contenido del capítulo en la página 33. Es posible que deba usar setwd para apuntar **R** a la carpeta donde ha guardado los datos. Después de esto, ejecute el siguiente código:

pres.energy<-read.csv("PESenergy.csv")  
  
hist(pres.energyEnergy,xlab="Television Stories",main="")  
  
abline(h=0,col=’gray60’)  
  
box()

El resultado que produce este código se presenta en la figura  3.1 . En este código, comenzamos leyendo Peake y Eshbaugh-Soha (2008) datos. El archivo de datos en sí es un archivo de valores separados por comas con una fila de encabezado de nombres de variables, por lo que los valores predeterminados de read.csv se adaptan a nuestros propósitos. Una vez que se cargan los datos, trazamos un histograma de nuestra variable de interés usando el comando hist: **pres.energy$Energy** llama a la variable de interés desde su marco de datos. Usamos la opción xlab, que nos permite definir la etiqueta que imprime **R** en el eje horizontal. Dado que este eje nos muestra los valores de la variable, simplemente deseamos ver la frase “Historias de televisión,” describiendo brevemente lo que significan estos números. El opción main define un título impreso en la parte superior de la figura. En este caso, la única forma de imponer un título en blanco es incluir citas sin contenido entre ellas. Una característica interesante de graficar en el paquete base es que algunos comandos pueden agregar información adicional a una gráfica que ya se ha dibujado. El comando abline es una herramienta flexible y útil. (El nombre **a-bline** se refiere a la fórmula lineal . Por lo tanto, este comando puede dibujar líneas con una pendiente y la intersección, o se puede dibujar una línea horizontal o vertical). En este caso, abline agrega un horizontal línea a lo largo del punto 0 en el eje vertical, por lo tanto . Esto se agrega para aclarar dónde está la base de las barras en la figura. Finalmente, el comando box() encierra la figura completa en un cuadro, a menudo útil en artículos impresos para aclarar dónde termina el espacio gráfico y comienza el otro espacio en blanco. Como muestra el histograma, hay una fuerte concentración de observaciones en 0 y justo por encima de 0, y un claro sesgo positivo en la distribución. (De hecho, estos datos se vuelven a analizar en Fogarty y Monogan (2014) precisamente para abordar algunas de estas características de datos y discutir los medios útiles de analizar los recuentos de medios dependientes del tiempo).

**Figura 3.1** Histograma del recuento mensual de noticias de televisión relacionadas con la energía

Otro gráfico univariado es un diagrama de caja y bigotes. **R** nos permite obtener esto únicamente para la variable única, o para un subconjunto de la variable basado en alguna otra medida disponible. Primero dibujando esto para una sola variable:

boxplot(pres.energy$Energy,ylab="Television Stories")

El resultado de esto se presenta en el panel (a) de la Fig.  3.2 . En este caso, los valores de los recuentos mensuales están en el eje vertical; por lo tanto, usamos la opción ylab para etiquetar el eje vertical (o el eje y **label**) apropiadamente. En la figura, la parte inferior del cuadro representa el valor del primer cuartil (percentil 25), la línea sólida grande dentro del cuadro representa el valor mediano (segundo cuartil, percentil 50) y la parte superior del cuadro representa el valor del tercer cuartil ( Percentil 75). Los bigotes, por defecto, se extienden a los valores más bajos y más altos de la variable que no son más de 1,5 veces el rango intercuartílico (o la diferencia entre el tercer y el primer cuartil) fuera de la caja. El propósito de los bigotes es transmitir el rango sobre el que cae la mayor parte de los datos. Los datos que quedan fuera de este rango se representan como puntos en sus valores respectivos. Esta gráfica de caja se ajusta a nuestra conclusión del histograma: los valores pequeños, incluido 0, son comunes y los datos tienen un sesgo positivo.

**Figura 3.2** Tramas de caja y bigotes de la distribución del recuento mensual de nuevas historias televisivas relacionadas con la energía. El panel ( a ) muestra la distribución completa y el panel ( b ) muestra las distribuciones para los subconjuntos antes y después de noviembre de 1973

Los diagramas de caja y bigotes también pueden servir para ofrecer una idea de la distribución condicional de una variable. Para nuestra serie temporal de cobertura de la política energética, el primer evento importante que observamos es el discurso de Nixon de noviembre de 1973 sobre el tema. Por lo tanto, podríamos crear un indicador simple donde los primeros 58 meses de la serie (hasta octubre de 1973) se codifican con 0 y los 122 meses restantes de la serie (desde noviembre de 1973 en adelante) se codifican con 1. Una vez que hacemos esto, el comando boxplot nos permite condicionar sobre una variable:

pres.energy$post.nixon<-c(rep(0,58),rep(1,122))  
  
boxplot(pres.energy$Energy~pres.energy$post.nixon,  
  
 axes=F,ylab="Television Stories")  
  
axis(1,at=c(1,2),labels=c("Before Nov. 1973",  
  
 "After Nov. 1973"))  
  
axis(2)  
  
box()

Esta salida se presenta en el panel (b) de la figura  3.2 . La primera línea de código define nuestra variable anterior a posterior a noviembre de 1973. Observe aquí que nuevamente definimos un vector con c. Dentro de c, usamos el comando rep (para **repeat**). Entonces rep(0,58) produce 58 ceros y rep(1,122) produce 122 unidades. La segunda línea dibuja nuestras gráficas de caja , pero agregamos dos advertencias importantes relativas a nuestra última llamada a la boxplot: Primero, enumeramos pres.energypost.nixon como nuestro argumento de datos. El argumento antes de la tilde (~) es la variable para la que queremos la distribución, y el argumento posterior es la variable condicionante. En segundo lugar, agregamos el comando axes = F. (También podríamos escribir axes = FALSE, pero **R** acepta F como abreviatura). Esto nos da más control sobre cómo se presentan los ejes horizontal y vertical. En el comando siguiente, agregamos el eje 1 (el eje horizontal inferior), agregando etiquetas de texto en las marcas de verificación 1 y 2 para describir los valores de la variable condicionante. Luego, agregamos el eje 2 (el eje vertical izquierdo) y un cuadro alrededor de toda la figura. Panel (b) de la figura  3.2muestra que la distribución antes y después de esta fecha es fundamentalmente diferente. Los valores mucho más pequeños persisten antes del discurso de Nixon, mientras que hay una media más grande y una mayor dispersión de valores después. Por supuesto, esto es solo un primer vistazo y el efecto del discurso de Nixon se confunde con una variedad de factores, como el precio del petróleo, la aprobación presidencial y la tasa de desempleo, que contribuyen a esta diferencia.

### Gráficos de barras

Los gráficos de barras pueden ser útiles siempre que queramos ilustrar el valor que toman algunas estadísticas para una variedad de grupos, así como para visualizar las proporciones relativas de datos medidos nominales u ordinalmente. Para ver un ejemplo de gráficos de barras, pasamos ahora al otro conjunto de datos de ejemplo de este capítulo, sobre el cabildeo por la salud en los 50 estados estadounidenses. Lowery y col. ofrecer un gráfico de barras de los promedios en todos los estados de la tasa de participación en el cabildeo, o el número de cabilderos como porcentaje del número de empresas, para todos los cabilderos de la salud y para siete subgrupos de cabilderos de la salud (2008, Fig. 3). Podemos recrear esa figura en **R** tomando las medias de estas ocho variables y luego aplicando la función de **barplot** al conjunto de medias. Primero debemos cargar los datos. Para hacer esto, descargue los datos de Lowery et al. Sobre cabildeo, el archivo llamado constructionData.dta . El archivo está disponible en el Dataverse nombrado en la página vii o en el enlace de contenido del capítulo en la página 33. De nuevo, es posible que deba usar setwd para apuntar **R** a la carpeta donde ha guardado los datos. Dado que estos datos están en formato Stata, debemos usar la biblioteca foreign y luego el comando read.dta:

library(foreign)  
  
health.fin<-read.dta("constructionData.dta")

Para crear la figura real en sí, podemos crear un subconjunto de nuestros datos que solo incluya los ocho predictores de interés y luego usar la función de apply para obtener la media de cada variable.

part.rates<-subset(health.fin,select=c(  
  
 partratetotalhealth,partratedpc,  
  
 partratepharmprod,partrateprofessionals,partrateadvo,  
  
 partratebusness,partrategov,rnmedschoolpartrate))  
  
lobby.means<-apply(part.rates,2,mean)  
  
names(lobby.means)<-c("Total Health Care",  
  
 "Direct Patient Care","Drugs/Health Products",  
  
 "Health Professionals","Health Advocacy","  
  
 Health Finance","Local Government","Health Education")

En este caso, part.rates es nuestro marco de datos subconjunto que solo incluye las ocho tasas de interés de participación del lobby. En la última línea, el comando **apply** permite tener un marco de matriz o de datos (part.rates) y aplicar una función de interés (mean) a cualquiera de las filas o las columnas de la trama de datos. Queremos la media de cada variable y las columnas de nuestro conjunto de datos representan las variables. El **2**, que es el segundo componente de este comando, le dice a **apply** que queremos aplicar la media a las *columnas* de nuestros datos. (Por el contrario, un argumento de **1** se aplicaría a las *filas*. Los cálculos basados ​​en filas serían útiles si necesitáramos calcular alguna cantidad nueva para cada uno de los 50 estados.) Si simplemente escribimos lobby.means en la consola **R** ahora, imprimirá los ocho medios de interés para nosotros. Para configurar nuestra cifra de antemano, podemos adjuntar un nombre en inglés a cada cantidad que se informará en el margen de nuestra cifra. Hacemos esto con el comando de names y luego asignamos un vector con un nombre para cada cantidad.

Para dibujar realmente nuestro gráfico de barras, usamos el siguiente código:

par(mar=c(5.1, 10 ,4.1 ,2.1))  
  
barplot(lobby.means,xlab="Percent Lobby Registration",  
  
 xlim=c(0,26),horiz=T,cex.names=.8,las=1)  
  
text(x=lobby.means,y=c(.75,1.75,3,4.25,5.5,6.75,8,9),  
  
 labels=paste(round(lobby.means,2)),pos=4)  
  
box()

Los resultados se representan en la figura  3.3 . La primera línea llama al comando **par**, que permite al usuario cambiar una amplia gama de valores predeterminados en el espacio gráfico. En nuestro caso, necesitamos un margen izquierdo más grande, por lo que usamos la opción **mar** para cambiar esto, estableciendo el segundo valor en el valor relativamente grande de 10. (En general, los márgenes se enumeran como inferior, izquierdo, superior y luego derecho .) Todo lo que se ajuste con par se restablece a los valores predeterminados después de cerrar la ventana de trazado (o dispositivo, si se escribe directamente en un archivo). A continuación, usamos el comando **barplot**. El argumento principal es lobby.means, que es el vector de medias variables. El valor predeterminado de **barplot** consiste en dibujar un gráfico con líneas verticales. En este caso, sin embargo, configuramos la opción horiz = T para obtener barras horizontales. También usamos las opciones cex.names (**c**haracter **ex**pansion for axis **names** o expansión de las caraterísticas en los nombres de los ejes) y las = 1 (**l**abel **a**xis **s**tyle) para reducir nuestras etiquetas de barras para el 80% de su tamaño por defecto y los obligan a imprimir en horizontal, respectivamente[[7]](#footnote-46). El comando **xlab** nos permite describir la variable para la cual estamos mostrando las medias, y el comando **xlim** (**x**-axis **lim**its) nos permite establecer el espacio de nuestro eje horizontal. Finalmente, usamos el comando **text** para imprimir la media de cada tasa de registro de lobby al final de la barra. El comando **text** es útil cada vez que deseamos agregar texto a un gráfico, ya sean valores numéricos o etiquetas de texto. Este comando toma coordenadas **x** para su posición a lo largo del eje horizontal, **y** coordenadas para su posición a lo largo del eje vertical y **labels** valores para que el texto se imprima en cada punto. La opción pos = 4 especifica imprimir el texto a la derecha del punto dado (alternativamente 1, 2 y 3 especificarían abajo, izquierda y arriba, respectivamente), para que nuestro texto no se superponga con la barra.

**Figura 3.3** Gráfico de barras de la tasa media de participación de los grupos de presión en la atención médica y en siete subgremios en los 50 estados de EE.

## La función **plot**

Pasamos ahora a **plot**, la función gráfica de caballo de batalla en el paquete **base**. El comando **plot** se presta naturalmente a gráficos bivariados. Para ver la suma total de argumentos que uno puede llamar usando **plot**, escriba args (plot.default) , que devuelve lo siguiente:

function (x, y=NULL, type="p", xlim=NULL, ylim=NULL,  
  
 log="", main=NULL, sub=NULL, xlab=NULL, ylab=NULL,  
  
 ann=par("ann"), axes=TRUE, frame.plot=axes,  
  
 panel.first=NULL, panel.last=NULL, asp=NA, ...)

Obviamente, están sucediendo muchas cosas debajo de la función **plot** genérica . Con el fin de comenzar con la creación de figuras en **R**  , queremos preguntarnos qué es esencial. La respuesta es sencilla: se debe especificar una variable **x**. Todo lo demás tiene un valor predeterminado o no es esencial. Para comenzar a experimentar con **plot**, continuamos utilizando los datos de cabildeo de salud del estado de 1997 cargados en la Sect. 3.1.1 .

Con **plot**, podemos trazar las variables por separado con el comando plot(varname), aunque esto es definitivamente menos informativo que los tipos de gráficos que se acaban de presentar en la Sección. 3.1 . Dicho esto, si simplemente quisiéramos ver todos los valores observados de la tasa de participación del lobby de las firmas financieras del estado de la salud (**partratebusness**), simplemente escribimos :

plot(health.fin$partratebusness,  
  
 ylab="Lobby Participation Rate")

La figura 3.4 a se devuelve en la interfaz gráfica R. Tenga en cuenta que esta figura traza la tasa de participación del lobby contra el número de fila en el marco de datos: con datos transversales, este índice es esencialmente insignificante. Por el contrario, si estuviéramos estudiando datos de series de tiempo y los datos se clasificaran a tiempo, podríamos observar cómo evoluciona la serie a lo largo del tiempo. Tenga en cuenta que usamos la opción **ylab** porque, de lo contrario, el valor predeterminado etiquetará nuestro eje vertical con el aspecto de mal gusto health.fin$partratebusness. (Pruébelo y pregúntese qué pensaría el editor de una revista sobre cómo se ve el resultado).

**Figura 3.4** Tasa de participación del lobby de la industria de las finanzas de la salud solo y contra el número de establecimientos comerciales de finanzas de la salud. ( a ) Índice. ( b ) Número de establecimientos de salud

Por supuesto, estamos más interesados ​​en las relaciones bivariadas. Podemos explorarlos fácilmente incorporando una variable x en el eje horizontal (generalmente *una variable independiente*) y una variable y en el eje vertical (generalmente *una variable dependiente*) en la llamada a graficar:

plot(y=health.fin$partratebusness,x=health.fin$supplybusiness,  
  
 ylab="Lobby Participation Rate",  
  
 xlab="Number of Health Establishments")

Esto produce la figura  3.4 b, donde nuestro eje horizontal se define por el número de empresas de financiación de la salud en un estado, y el eje vertical se define por la tasa de participación del lobby de estas empresas en el estado respectivo. Este gráfico muestra lo que parece ser una disminución en la tasa de participación a medida que aumenta el número de empresas, quizás en una relación curvilínea.

Una herramienta útil es trazar la forma funcional de un modelo bivariado en el diagrama de dispersión de las dos variables. En el caso de la figura  3.4 b, es posible que deseemos comparar cómo una función lineal versus una función cuadrática o al cuadrado del número de empresas se ajusta al resultado de la tasa de participación del lobby. Para hacer esto, podemos ajustar dos modelos de regresión lineal, uno que incluye una función lineal de número de empresas y el otro que incluye una función cuadrática. Los detalles adicionales sobre los modelos de regresión se analizan más adelante en el Cap. 6 Nuestros dos modelos en este caso son:

finance.linear<-lm(partratebusness~supplybusiness,  
  
 data=health.fin)  
  
summary(finance.linear)  
  
finance.quadratic<-lm(partratebusness~supplybusiness+  
  
 I(supplybusiness^2),data=health.fin)  
  
summary(finance.quadratic)

El comando **lm** (**l**inear **m**odel ) se ajusta a nuestros modelos y el comando summary resume nuestros resultados. Nuevamente, los detalles de lm se discutirán en el cap. 6 Con el modelo que es una función lineal del número de empresas, simplemente podemos ingresar el nombre de nuestro modelo ajustado (finance.linear) en el comando abline para agregar nuestra línea de regresión ajustada al gráfico:

plot(y=health.fin$partratebusness,x=health.fin$supplybusiness,  
  
 ylab="Lobby Participation Rate",  
  
 xlab="Number of Health Establishments")  
  
abline(finance.linear)

Como se mencionó anteriormente, el comando **abline** es particularmente flexible. Un usuario puede especificar **a** como la intersección de una línea y **b** como la pendiente. Un usuario puede especificar **h** como el valor del eje vertical donde se dibuja una línea horizontal, o v como el valor del eje horizontal donde se dibuja una línea vertical. O, en este caso, se puede insertar un modelo de regresión con un predictor para dibujar la línea de regresión que mejor se ajuste. Los resultados se presentan en la figura  3.5 a.

**Figura 3.5** Tasa de participación del lobby de la industria financiera de la salud contra el número de establecimientos de salud, modelos lineales y cuadráticos. ( a ) Función lineal. ( b ) Función cuadrática

Alternativamente, podríamos volver a dibujar este gráfico con la relación cuadrática esbozada en él. Desafortunadamente, a pesar de la flexibilidad de **abline**, no puede trazar una relación cuadrática por defecto. La forma más fácil de trazar una forma funcional compleja es guardar los valores predichos del modelo, reordenar los datos según el predictor de interés y luego usar la función **lines** para agregar una línea conectada de todas las predicciones. Asegúrese de que los datos estén ordenados correctamente en el predictor; de lo contrario, la línea aparecerá como un desorden. El código en este caso es:

plot(y=health.fin$partratebusness,x=health.fin$supplybusiness,  
  
 ylab="Lobby Participation Rate",  
  
 xlab="Number of Health Establishments")  
  
finance.quadratic<-lm(partratebusness~supplybusiness+  
  
 I(supplybusiness^2), data=health.fin)  
  
health.fin$quad.fit<-finance.quadratic$fitted.values  
  
health.fin<-health.fin[order(health.fin$supplybusiness),]  
  
lines(y=health.fin$quad.fit,x=health.fin$supplybusiness)

Este resultado se presenta en la figura  3.5 b. Si bien aún no entraremos en detalles de **lm**, observe que I(supplybusiness^2) se usa como predictor. **I** significa “como **i**s,” por lo que nos permite calcular una fórmula matemática sobre la marcha. Después de volver a dibujar nuestro diagrama de dispersión original, estimamos nuestro modelo cuadrático y guardamos los valores ajustados en nuestro marco de datos como la variable quad.fit. En la cuarta línea, reordenamos nuestro marco de datos health.fin de acuerdo con los valores de nuestra variable de entrada supplybusiness. Esto se hace mediante el comando **order**, que enumera los índices vectoriales en orden de valor creciente. Finalmente, el comando **lines** toma nuestros valores predichos como las coordenadas verticales (**y**) y nuestros valores del número de empresas como las coordenadas horizontales (**x**). Esto agrega la línea al gráfico que muestra nuestra forma funcional cuadrática.

### Gráficos lineales con **plot**

Hasta ahora, nuestros análisis se han basado en el **plot** predeterminado de dibujar un diagrama de dispersión. Sin embargo, en el análisis de series de tiempo, un gráfico de líneas a lo largo del tiempo suele ser útil para observar las propiedades de la serie y cómo cambia con el tiempo. (Más información sobre esto está disponible en el Capítulo  9 ) Volviendo a los datos sobre la cobertura de noticias televisivas de la política energética planteados por primera vez en la Sect. 3.1 , visualicemos el resultado de la cobertura de la política energética y un insumo del precio del petróleo.

Comenzando con la cantidad de historias de energía por mes, creamos esta trama de la siguiente manera:

plot(x=pres.energy$Energy,type="l",axes=F,  
  
 xlab="Month", ylab="Television Stories on Energy")  
  
axis(1,at=c(1,37,73,109,145),labels=c("Jan. 1969",  
  
 "Jan. 1972","Jan. 1975","Jan. 1978","Jan. 1981"),  
  
 cex.axis=.7)  
  
axis(2)  
  
abline(h=0,col="gray60")  
  
box()

Esto produce la figura  3.6 a. En este caso, nuestros datos ya están ordenados por mes, por lo que si solo especificamos **x** sin **y**, **R** mostrará todos los valores en el orden temporal correcto[[8]](#footnote-49). ara designar que queremos un diagrama de líneas en lugar de un diagrama de dispersión de puntos, insertamos la letra **l** en la opción **type = “l”**. En este caso, hemos desactivado los ejes porque las marcas de graduación predeterminadas para el mes no son particularmente significativas. En su lugar, usamos el comando **axis** para insertar una etiqueta para el primer mes del año cada 3 años, ofreciendo una mejor sensación del tiempo real. Observe que en nuestra primera llamada a **axis**, usamos la opción **cex.axis** para encoger nuestras etiquetas al 70% del tamaño. Esto permite que las cinco etiquetas quepan en el gráfico. (Por ensayo y error, verá que **R** deja caer las etiquetas del eje que no encajarán en lugar de sobreimprimir el texto). Finalmente, usamos **abline** para mostrar el punto cero en el eje vertical, ya que este es un número significativo que refleja la ausencia completa. de la cobertura de la política energética en los informativos televisivos. Como demostraron nuestras cifras anteriores, vemos mucha más variabilidad y una media más alta después de los primeros 4 años. La cifra del precio del petróleo por barril se puede crear de manera similar:

**Figura 3.6** Número de reportajes televisivos sobre política energética y precio del petróleo por barril, respectivamente, por mes. ( a

plot(x=pres.energy$oilc,type="l",axes=F,xlab="Month",  
  
 ylab="Cost of Oil")  
  
axis(1,at=c(1,37,73,109,145),labels=c("Jan. 1969",  
  
 "Jan. 1972","Jan. 1975","Jan. 1978","Jan. 1981"),  
  
 cex.axis=.7)  
  
axis(2)  
  
box()

Nuevamente, los datos están ordenados, por lo que solo se necesita una variable. La figura 3.6 b presenta este gráfico.

### Construcción de figuras con **plot**: Detalles adicionales

Después de haber probado la mano con parcelas de la **base** de paquete, ahora itemize en detalle las funciones y opciones que aportan una considerable flexibilidad para crear figuras básicas en **R** . Tenga en cuenta que **R** en realidad ofrece la opción útil de comenzar con una pizarra en blanco y agregar elementos al gráfico bit a bit.

**El sistema de coordenadas:**

En la figura  3.4 , no nos preocupó establecer el sistema de coordenadas porque los datos efectivamente lo hicieron por nosotros. Pero a menudo, querrá establecer las dimensiones de la figura antes de trazar cualquier cosa, especialmente si está construyendo a partir del lienzo en blanco. El punto más importante aquí es que sus x e y deben ser de la misma longitud. Esto quizás sea obvio, pero los datos faltantes pueden crear dificultades que llevarán a **R** a oponerse.

**Tipos de gráficos:**

Ahora queremos graficar estas series, pero la función de trazado permite diferentes tipos de gráficos. Los diferentes tipos que se pueden incluir dentro de la función de gráfico genérico incluyen:

* **type = “p”** Esto es el valor predeterminado y grafica la **x** y **y** coordenadas como *puntos*.
* **type = “l”** Esto grafica la **x** y **y** coordenadas como *líneas*.
* **type = “n”** Esto grafica la **x** y **y** coordenadas como *nada* (solo configura el espacio de coordenadas).
* **type = “O”** Esto grafica la **x** y **y** coordenadas como *puntos y líneas* superpuestas (es decir, “overplots”).
* **type = “h”** Esto grafica la **x** y **y** coordenadas como un *histograma como líneas verticales* (también llamado *diagrama de picos*).
* **type = “s”** Esto grafica la **x** y **y** coordenadas como en *pasos de escalera como líneas*.

**Ejes:**

Es posible apagar los ejes, ajustar el espacio de coordenadas usando las opciones **xlim** e **ylim**, y crear sus propias etiquetas para los ejes. - **axes =** Le permite controlar si los ejes aparecen en la figura o no. Si tiene fuertes preferencias sobre cómo se crean sus ejes, puede desactivarlos seleccionando **axes = F** dentro de **plot** y luego cree sus propias etiquetas usando el comando de **axis** separado : axis(side=1,at=c(2,4,6,8,10,12),labels=c(“Feb,” “Apr,”“June,”“Aug,”“Oct,”“Dec”)) - **xlim =**, **ylim =** Por ejemplo, si quisiéramos expandir el espacio desde el valor predeterminado de **R** , podríamos ingresar: plot(x=ind.var, y=dep.var, type=“o,” xlim=c(-5, 17),ylim=c(-5, 15)) - **xlab = "“, ylab =”"** Crea etiquetas para los ejes **x** e **y**.

**Estilo:**

Hay una serie de opciones para ajustar el estilo en la figura, incluidos cambios en el tipo de línea, grosor de línea, color, estilo de punto y más. Algunos comandos comunes incluyen: - **asp =** Define el relación de **asp**ecto de la gráfica. Establecer **asp = 1** es una opción poderosa y útil que permite al usuario declarar que los dos ejes se miden en la misma escala. Consulte la Fig.  5.1 en la página 80 y la Fig.  8.4 en la página 159 como dos ejemplos de esta opción.  
- **lty =** Selecciona el tipo de línea (sólida, discontinua, guión corto-largo, etc.). - **lwd =** Selecciona el ancho de la línea (líneas gruesas o delgadas). - **pch =** Selecciona el símbolo de trazado, puede ser un símbolo numerado (**pch = 1**) o una letra (**pch = “D”**). - **col =** Selecciona el color de las líneas o puntos de la figura. - **cex =** Factor de **C**haracter **ex**pansion (expansión de caracter) que ajusta el tamaño del texto y los símbolos en la figura. De manera similar, **cex.axis** ajusta el tamaño de la anotación del eje, **cex.lab** ajusta el tamaño de la fuente para las etiquetas del eje, **cex.main** ajusta el tamaño de la fuente del título y **cex.sub** ajusta el tamaño de la fuente de los subtítulos.

**Parámetros gráficos:**

La función **par** trae añadió funcionalidad para el trazado en **R** , dando el control del usuario sobre las representación gráfica (**par**ameters). Una característica notable de **par** es que le permite trazar múltiples llamadas de **plot** en un solo gráfico. Esto se logra seleccionando **par(new = T)**, mientras una ventana de trazado (o dispositivo) todavía está abierta y antes de la siguiente llamada a **plot**. Pero *ten cuidado*. Cada vez que use esta estrategia, incluya los comandos **xlim** e **ylim** en cada llamada para asegurarse de que el espacio de la gráfica se mantenga igual. También tenga cuidado de que los márgenes del gráfico no cambien de una llamada a la siguiente.

### Funciones complementarias

También hay una serie de funciones complementarias que se pueden utilizar una vez que se ha creado el sistema de coordenadas básico mediante **plot**. Éstas incluyen: - **arrows(x1, y1, x2, y2)** Cree flechas dentro de la gráfica (útil para etiquetar puntos de datos particulares, series, etc.). - **text(x1, x2, “text”)** Cree texto dentro de la trama (modifique el tamaño del texto usando la opción de expansión de caracteres **cex**). - **lineas (x, y)** Crea una gráfica que conecte líneas. - **puntos (x, y)** Crea una gráfica de puntos. - **polígono()** Crea un polígono de cualquier forma (rectángulos, triángulos, etc.). - **legend(x, y, at = c("“,”“), labels=c(”“,”"))** Cree una leyenda para identificar los componentes de la figura. - **axis(side)** Agregue un eje con etiquetas predeterminadas o personalizadas a uno de los lados de un gráfico. Establezca el lado en 1 para la parte inferior, 2 para la izquierda, 3 para la parte superior y 4 para la derecha. - **mtext(text, side)** Comando para agregar **m**argen de **text**o. Esto le permite agregar una etiqueta de eje a uno de los lados con más control sobre cómo se presenta la etiqueta. Consulte el código que produce la Fig.  7.1 en la página 114 para ver un ejemplo de esto.

## Usar gráficos **lattice** en R

Como alternativa al paquete de gráficos base , es posible que desee considerar el paquete de complementos de celosía . Estos producen gráficos de enrejado a partir del lenguaje S , que tienden a mostrar mejor los datos agrupados y numerosas observaciones. Algunas características interesantes del paquete lattice son que los gráficos tienen valores predeterminados fáciles de usar y los comandos ofrecen una opción data = que no requiere que el usuario enumere el marco de datos con cada llamada a una variable.

Para empezar, la primera vez que usamos la biblioteca lattice , debemos instalarla. Luego, en cada reutilización del paquete, debemos llamarlo con el comando de la biblioteca .

install.packages (’’ celosía ’’)

biblioteca (celosía)

Para obtener una gráfica de dispersión similar a la que dibujamos con plot , esto se puede lograr en celosía usando el comando xyplot :

xyplot (partratebusness ~ supplybusiness, data = health.fin,

col = "black", ylab = "Tasa de participación en el lobby",  
  
 xlab = "Número de establecimientos de salud")

La Figura 3.7 a muestra este gráfico. La sintaxis difiere un poco de la función plot : en este caso, podemos especificar una opción, data = health.fin , que nos permite escribir el nombre del marco de datos relevante una vez, en lugar de volver a escribirlo para cada variable. Además, ambas variables se enumeran juntas en un solo argumento usando la forma, vertical.variable ~ horizontal.variable . En este caso, también especificamos la opción, col = “negro” con el fin de producir una figura en blanco y negro. Por defecto, los colores de celosía dan como resultado cian para permitir a los lectores separar fácilmente la información de datos de otros aspectos de la pantalla, como ejes y etiquetas (Becker et al. 1996, pag. 153). Además, de forma predeterminada, xyplot imprime marcas de verificación en el tercer y cuarto eje para proporcionar puntos de referencia adicionales para el espectador. Abrir imagen en nueva ventanaFigura 3.7 Figura 3.7 Tasa de participación en el lobby de la industria de las finanzas de la salud en comparación con el número de establecimientos de salud, ( a ) diagrama de dispersión y ( b ) diagrama de puntos

El paquete de celosía también contiene funciones que dibujan gráficos que son similares a un diagrama de dispersión, pero en su lugar utilizan un orden de clasificación de la variable del eje vertical. Así es como funcionan los comandos stripplot y dotplot , y ofrecen otra vista de una relación y su solidez. El comando dotplot puede ser algo más deseable, ya que también muestra una línea para cada valor ordenado por rango, lo que da la sensación de que la escala es diferente. La sintaxis de la gráfica de puntos se ve así:

dotplot (partratebusness ~ supplybusiness,

datos = salud.fin, col = "negro",  
  
 ylab = "Tasa de participación en el lobby (orden de clasificación)",  
  
 xlab = "Número de establecimientos de salud")

La figura 3.7 b muestra este resultado. La función stripplot usa una sintaxis similar.

Por último, la biblioteca de celosía nuevamente nos da la opción de ver la distribución de una sola variable trazando un histograma o un gráfico de densidad. Volviendo a los datos de la serie temporal presidencial que cargamos por primera vez en la Secta. 3.1 , ahora podemos dibujar una gráfica de densidad usando la siguiente línea de código:

diagrama de densidad (~ Energía, datos = energía pres.,

xlab = "Historias de televisión", col = "negro")

Esto se presenta en la figura  3.8 a. Esta salida muestra puntos dispersos a lo largo de la base, cada uno de los cuales representa el valor de una observación. La línea suavizada a lo largo del gráfico representa la densidad relativa estimada de los valores de la variable. Abrir imagen en nueva ventanaFigura 3.8 Figura 3.8 ( a ) Gráfico de densidad e ( b ) histograma que muestra la distribución univariante del recuento mensual de noticias de televisión relacionadas con la energía

Alternativamente, se puede dibujar un histograma en celosía con la función histograma :

histograma (~ Energía, datos = energía pres.,

xlab = "Historias de televisión", col = "gray60")

Esto está impreso en la Fig.  3.8 b. En este caso, el color se establece en col = “gray60” . De nuevo, el valor predeterminado es para barras de color cian. Para una buena opción de escala de grises en este caso, un gris medio aún permite distinguir claramente cada barra. Una última característica interesante del histograma se deja al lector: la función dibujará distribuciones de histograma condicionales. Si todavía tiene disponible la variable post.nixon que creamos anteriormente, puede intentar escribir histograma (~ Energy | post.nixon, data = pres.energy) , donde la tubería vertical ( | ) es seguida por la variable condicionante.

3.4 Salida gráfica Un último punto esencial es una palabra sobre cómo los usuarios pueden exportar sus gráficos **R** a un procesador de texto o editor de escritorio deseado. La primera opción es guardar la salida de pantalla de una figura. En máquinas Mac, el usuario puede seleccionar la ventana de salida de la figura y luego usar el menú desplegable Fi l e → Sa v e A s … ——————- para guardar la figura como un archivo PDF. En máquinas con Windows, un usuario puede simplemente hacer clic con el botón derecho en la ventana de salida de la figura y luego elegir guardar la figura como un metarchivo (que se puede usar en programas como Word) o como un archivo postscript (para usar en LaTeX) . También al hacer clic con el botón derecho en Windows, los usuarios pueden copiar la imagen y pegarla en Word, PowerPoint o un programa de gráficos.

Una segunda opción permite a los usuarios una mayor precisión sobre el producto final. Específicamente, el usuario puede escribir el gráfico en un dispositivo gráfico, del cual hay varias opciones. Por ejemplo, al escribir este libro, exporté la Fig.  3.5 a escribiendo:

postscript (“lin.partrate.eps,” horizontal = FALSE, ancho = 3,

altura = 3, un archivo = FALSO, papel = "especial", tamaño de puntos = 7)

plot (y = health.fin $ partratebusness, x = health.fin $ supplybusiness,

ylab = "Tasa de participación en el lobby",  
  
 xlab = "Número de establecimientos de salud")

abline (finanzas.linear)

dev.off ()

La primera línea llama al comando postscript , que creó un archivo llamado lin.partrate.eps en el que guardé el gráfico. Entre las opciones clave de este comando se encuentran el ancho y el alto , cada uno de los cuales configuré en tres pulgadas. El comando de tamaño de puntos redujo el texto y los símbolos para encajar perfectamente en el espacio que asigné. El comando horizontal cambia la orientación del gráfico de horizontal a vertical en la página. Cámbielo a TRUE para que el gráfico adopte una orientación horizontal. Una vez que se llamó a postscript , todos los comandos de gráficos se escribieron en el archivo y no en la ventana de gráficos.. Por lo tanto, suele ser una buena idea perfeccionar un gráfico antes de escribirlo en un dispositivo gráfico. Por lo tanto, los comandos plot y abline sirvieron para escribir toda la salida en el archivo. Una vez que terminé de escribir en el archivo, el comando dev.off () cerró el archivo para que ningún otro comando de gráficos escribiera en él.

Por supuesto, los escritores que utilizan el lenguaje de autoedición de LaTeX utilizan con mayor frecuencia los gráficos postscript. Los escritores que utilizan procesadores de texto más tradicionales, como Word o Pages, querrán utilizar otros dispositivos gráficos. Las opciones disponibles incluyen: jpeg , pdf , png y tiff . 5 Para utilizar cualquiera de estos cuatro dispositivos gráficos, sustituya una llamada por la función relevante donde se encuentra la posdata en el código anterior. Sin embargo, asegúrese de escribir ? Png para familiarizarse con la sintaxis de estos dispositivos alternativos, ya que cada uno de los cinco tiene una sintaxis ligeramente diferente.

Como circunstancia especial, los gráficos extraídos del paquete lattice utilizan un dispositivo gráfico diferente, llamado trellis.device . Es técnicamente posible utilizar los otros dispositivos gráficos para escribir en un archivo, pero no es aconsejable porque las opciones del dispositivo (por ejemplo, el tamaño del gráfico o el tamaño de la fuente) no se pasarán al gráfico. En el caso de la Fig.  3.7 b, generé la salida usando el siguiente código:

trellis.device (“postscript,” file = “dotplot.partrate.eps,”

tema = lista (tamaño de fuente = lista (texto = 7, puntos = 7)),  
  
 horizontal = FALSO, ancho = 3, alto = 3,  
  
 onefile = FALSE, paper = "especial")

dotplot (partratebusness ~ supplybusiness,

datos = salud.fin, col = 'negro',  
  
 ylab = "Tasa de participación en el lobby (orden de clasificación)",  
  
 xlab = "Número de establecimientos de salud")

dev.off ()

El primer argumento del comando trellis.device declara qué controlador desea utilizar el autor. Además de la posdata , el autor puede usar jpeg , pdf o png . El segundo argumento enumera el archivo en el que escribir. El tamaño de fuente y carácter debe establecerse a través de la opción de tema , y los argumentos restantes declaran las otras preferencias sobre la salida.

En este capítulo se ha cubierto funciones bivariadas de graficar en el **R** . Se han abordado varios comandos de los paquetes base y lattice . Esto está lejos de ser una lista exhaustiva de las capacidades gráficas de **R** , y se anima a los usuarios a aprender más sobre las opciones disponibles. Este cebador debe, sin embargo, sirven para introducir a los usuarios a diversos medios por los que los datos pueden ser visualizados en **R** . Con un buen sentido de cómo tener una idea visual de los atributos de nuestros datos, el siguiente capítulo se centra en resúmenes numéricos de nuestros datos recopilados a través de estadísticas descriptivas.

## Problemas de práctica

Además de su análisis de la cobertura de la política energética presentado en este capítulo, Peake y Eshbaugh-Soha (2008) también estudian la cobertura de la póliza de medicamentos. Estos datos cuentan de manera similar el número de noticias de televisión nocturnas en un mes centradas en las drogas, desde enero de 1977 hasta diciembre de 1992. Sus datos se guardan en formato separado por comas en el archivo llamado drugCoverage.csv . Descargue sus datos del Dataverse mencionado en la página vii o del enlace de contenido del capítulo en la página 33. Las variables en este conjunto de datos son: un índice de tiempo basado en caracteres que muestra el mes y el año ( año ), cobertura de noticias de drogas ( drugsmedia ), un indicador de un discurso sobre drogas que pronunció Ronald Reagan en septiembre de 1986 ( rwr86 ), un indicador de un discurso que pronunció George HW Bush en septiembre de 1989 ( ghwb89 ), índice de aprobación del presidente ( aprobación) y la tasa de desempleo ( desempleo ).

1. Dibuja un histograma del recuento mensual de historias relacionadas con las drogas. Puede utilizar cualquiera de los comandos de histograma descritos en el capítulo.
2. Dibuja dos diagramas de caja: uno de historias relacionadas con las drogas y otro de aprobación presidencial. ¿En qué se diferencian estas cifras y qué le dice eso sobre el contraste entre las variables?
3. Dibuja dos diagramas de dispersión:
   * 1. En el primero, represente el número de historias relacionadas con las drogas en el eje vertical y coloque la tasa de desempleo en el eje horizontal.
     2. En el segundo, represente el número de historias relacionadas con las drogas en el eje vertical y coloque la aprobación presidencial en el eje horizontal.
     3. ¿En qué se diferencian las gráficas? ¿Qué te dicen de los datos?
     4. Bonificación: agregue una línea de regresión lineal a cada uno de los diagramas de dispersión.
4. Dibuja dos gráficos de líneas:
   * 1. En el primero, dibuje el número de historias relacionadas con las drogas por mes a lo largo del tiempo.
     2. En el segundo, obtenga la aprobación presidencial por mes a lo largo del tiempo.
     3. ¿Qué puedes aprender de estos gráficos?
5. Cargue la biblioteca de celosía y dibuje una gráfica de densidad del número de historias relacionadas con drogas por mes.
6. Bonificación: Dibuje un gráfico de barras de la frecuencia de las tasas de desempleo observadas. ( Sugerencia: intente usar el comando de tabla para crear el objeto que graficará). ¿Puede ir un paso más allá y dibujar una gráfica de barras del porcentaje de tiempo que se observa cada valor?

## References

* Becker RA, Cleveland WS, Shyu M-J (1996) The visual design and control of Trellis display. J Comput Graph Stat 5(2):123–155
* Beniger JR, Robyn DL (1978) Quantitative graphics in statistics: a brief history. Am Stat 32(1):1–11
* Chang W (2013) R graphics cookbook. O’Reilly, Sebastopol, CA
* Cleveland WS (1993) Visualizing data. Hobart Press, Sebastopol, CA
* Fogarty BJ, Monogan JE III (2014) Modeling time-series count data: the unique challenges facing political communication studies. Soc Sci Res 45:73–88
* Halley E (1686) An historical account of the trade winds, and monsoons, observable in the seas between and near the tropicks, with an attempt to assign the phisical cause of the said winds. Philos Trans 16(183):153–168
* Kastellec JP, Leoni EL (2007) Using graphs instead of tables in political science. Perspect Polit 5(4):755–771
* Lowery D, Gray V, Monogan JE III (2008) The construction of interest communities: distinguishing bottom-up and top-down models. J Polit 70(4):1160–1176
* Peake JS, Eshbaugh-Soha M (2008) The agenda-setting impact of major presidential TV addresses. Polit Commun 25:113–137
* Playfair W (1786/2005) In: Wainer H, Spence I (eds) Commercial and political atlas and statistical breviary. Cambridge University Press, New York
* Tufte ER (2001) The visual display of quantitative information, 2nd edn. Graphics Press, Cheshire, CT
* Tukey JW (1977) Exploratory data analysis. Addison-Wesley, Reading, PA
* Yau N (2011) Visualize this: the FlowingData guide to design, visualization, and statistics. Wiley, Indianapolis

# 4 Estadísticas descriptivas

Palabras clave: - Variable del indicador - Tendencia central - Marco de datos - Tabla de frecuencia - Desviación absoluta mediana

Antes de desarrollar cualquier modelo con un conjunto de datos o intentar extraer alguna inferencia a partir de un conjunto de datos, el usuario primero debe tener una idea de las características de los datos. Esto se puede lograr mediante los métodos de visualización de datos descritos en el Cap. 3 , así como a través de estadísticas descriptivas de la tendencia central y la dispersión de una variable, descritas en este capítulo. Idealmente, el usuario realizará ambas tareas, independientemente de si los resultados se convierten en parte del producto final publicado. Una recomendación tradicional para los analistas que estiman funciones como los modelos de regresión es que la primera tabla del artículo debe describir las estadísticas descriptivas de todas las variables de entrada y la variable de resultado. Si bien algunas revistas han dejado de utilizar el escaso espacio impreso en las tablas de estadísticas descriptivas, un buen analista de datos siempre creará esta tabla por sí mismo. Con frecuencia, esta información puede al menos incluirse en apéndices en línea, si no en la versión impresa del artículo.

A medida que trabajamos con estadísticas descriptivas, el ejemplo de trabajo en este capítulo serán datos centrados en políticas de LaLonde (1986) análisis de la Demostración Nacional de Trabajo Apoyado, un programa de la década de 1970 que ayudó a los desempleados de larga duración a encontrar trabajos en el sector privado y cubrió los costos laborales de su empleo durante un año. Las variables en este marco de datos son: - **treated:** Variable indicadora de si el participante recibió el tratamiento. - **age:** Medido en años. - **education:** Años de educación. - **black:** Variable indicadora de si el participante es afroamericano. - **married:** Variable indicadora de si el participante está casado. - **nodegree:** Variable indicadora de no poseer título de bachillerato. - **re74:** Ganancias reales en 1974. - **re75:** Ganancias reales en 1975. - **re78:** Ganancias reales en 1978. - **hispanic:** Variable indicadora de si el participante es hispano. - **u74:** Variable indicadora de desempleados en 1974. - **u75:** Variable indicadora de desempleados en 1975.

## Medidas de tendencia central

Nuestra primera tarea será calcular las medidas de centralidad, que nos dan una idea de un valor típico de una distribución. Las medidas más comunes de tendencia central son la media, la mediana y la moda. El rango intercuartil, que ofrece el 50% medio de los datos, también es informativo. Para comenzar con algunos cálculos de ejemplo, primero debemos cargar los datos de LaLonde (llamados **LL**). Estos están disponibles como parte del paquete **C**oarsened **E**xact **M**atching (**cem**), del que hablaremos con mayor detalle en el Cap. 8 Al igual que con cualquier otro paquete definido por el usuario, nuestra primera tarea es instalar el paquete:

install.packages("cem")  
  
library(cem)  
  
data(LL)

Después de instalar el paquete, cargamos la biblioteca, como tendremos que hacer en cada sesión en la que usemos el paquete. Una vez que se carga la biblioteca, podemos cargar los datos simplemente llamando al comando **data**, que carga este marco de datos guardado del paquete **cem** en la memoria de trabajo. Convenientemente podemos referirnos al marco de datos con el nombre **LL**^{Estos datos también están disponibles en formato separado por comas en el archivo llamado LL.csv. Este archivo de datos se puede descargar del Dataverse en la página vii o del enlace de contenido del capítulo en la página 53.}.

Para todas las medidas de tendencia central que calculamos, supongamos que tenemos una sola variable , con valores diferentes: . También podríamos ordenar los valores de menor a mayor, que se designa de manera diferente con las *estadísticas* de orden como: . En otras palabras, si alguien le preguntara por la estadística de segundo orden, le diría el valor de , el segundo valor más pequeño de la variable.

Con una variable como esta, la medida de centralidad más utilizada es la media muestral . Matemáticamente, calculamos esto como el promedio de los valores observados:

Dentro de **R**, podemos aplicar la Ec.(4.1) utilizando la función mean. Entonces, si x en este caso fueran los ingresos que los participantes en la Demostración Nacional de Trabajo Apoyado ganaron en 1974, aplicaríamos la función a la variable re74:

mean(LL$re74)

**R** responde imprimiendo [1] 3630.738, por lo que podemos informar la media de la muestra como .

Por supuesto, es recomendable seguir visualizando los datos utilizando las herramientas del Cap. 3 . Además de calcular la media de las ganancias reales en 1974, también podemos aprender mucho simplemente dibujando una gráfica de densidad. Podríamos hacer esto usando el código de **lattice** descrito en el último capítulo o con un poco más de control de usuario de la siguiente manera:

dens.74<-density(LL$re74,from=0)  
  
plot(dens.74,main="Income in 1974")  
  
abline(v=mean(LL$re74),col="red")

En la primera línea, el comando density nos permite calcular la densidad de observaciones en cada valor de ingreso. Con la opción from, podemos especificar que el valor mínimo posible de ingresos es 0 (y la opción to nos habría permitido establecer un máximo). En la segunda línea, simplemente trazamos este objeto de densidad. Por último, usamos abline para agregar una línea vertical donde se encuentra nuestra media calculada de $ 3,630.74.

El gráfico resultante se muestra en la Fig.  4.1 . Esta cifra es reveladora: la mayor parte de los datos están por debajo de la media. La media es tan alta porque un puñado de ingresos muy grandes (que se muestran en la larga cola derecha del gráfico) la están elevando. Con la imagen, rápidamente tenemos una idea de la distribución general de los datos.

**Figura 4.1** Gráfico de densidad de los ingresos reales en 1974 a partir de los datos de la demostración nacional de trabajo con apoyo

Volviendo a las representaciones estadísticas, otra medida común de tendencia central es la *mediana* muestral. Una ventaja de calcular una mediana es que es más robusta a valores extremos que la media. Imagínese si nuestra muestra hubiera incluido de alguna manera a Warren Buffett; nuestra estimación del ingreso medio habría aumentado sustancialmente con una sola observación. La mediana, por el contrario, se movería muy poco en respuesta a una observación tan extrema. Nuestra fórmula para calcular una mediana con datos observados se convierte en las estadísticas de orden que definimos anteriormente:

Tenga en cuenta que la notación para la mediana está algo dispersa y es uno de los varios símbolos de uso común. Formalmente, siempre que tenemos un número impar de valores, simplemente tomamos la estadística de orden medio (o el valor medio cuando los datos se ordenan de menor a mayor). Siempre que tengamos un número par de valores, tomamos las dos estadísticas de orden medio y promediamos entre ellas. (Por ejemplo, para las observaciones de diez, divide la diferencia entre y para obtener la mediana.) **R** ordenará nuestros datos, encontrar los valores de en medio, y tomar las medias reportar la mediana si simplemente Tipo:

median(LL$re74)

En este caso, **R** imprime [1] 823.8215, por lo que podemos informar como el ingreso medio para los participantes del programa en 1974. Observe que el valor medio es mucho más bajo que el valor medio, $ 2,806.92 más bajo, para ser exactos. Esto es consistente con lo que vimos en la Fig.  4.1 : Tenemos un sesgo positivo en nuestros datos, con algunos valores extremos que elevan un poco la media. Más adelante, lo verificaremos aún más observando los cuantiles de nuestra distribución.

Una tercera medida útil de tendencia central informa un rango de valores centrales. El rango intercuartil es el 50% medio de los datos. Usando estadísticas de pedidos, calculamos los límites inferior y superior de esta cantidad como:

Las dos cantidades informadas se denominan primer y tercer cuartiles . El primer cuartil es un valor para el cual el 25% de los datos son menores o iguales al valor. Del mismo modo, el 75% de los datos son menores o iguales al tercer cuartil. De esta manera, el 50% medio de los datos cae entre estos dos valores. En **R**, hay dos comandos que podemos escribir para obtener información sobre el rango intercuartil:

summary(LL$re74)  
  
IQR(LL$re74)

El comando de resumen es útil porque presenta la mediana y la media de la variable en un lugar, junto con el mínimo, el máximo, el primer cuartil y el tercer cuartil. Nuestra salida de **R**se ve así:

Min. 1er Qu. Mediana Media 3er Qu. Max.

0.0 0.0 823.8 3631.0 5212.0 39570.0

Esto es útil para obtener tres medidas de tendencia central a la vez, aunque tenga en cuenta que los valores de la media y la mediana de forma predeterminada se redondean a menos dígitos de los que informaron los comandos separados. Mientras tanto, el rango intercuartílico se puede leer en la salida impresa como IQR x  = [0, 5212]. Normalmente, diríamos que al menos el 50% de los participantes tenía un ingreso entre $ 0 y $ 5212. En este caso, sin embargo, sabemos que nadie obtuvo un ingreso negativo, por lo que el 75% de los encuestados cayó en este rango. Finalmente, el comando IQR reporta la diferencia entre el tercer y primer cuartil, en este caso imprimiendo: [1] 5211.795 . Este comando, entonces, simplemente informa la dispersión entre la parte inferior y superior del rango intercuartílico, nuevamente con menos redondeo que el que habríamos obtenido al usar los números informados por el resumen .

En la mayoría de las circunstancias, el redondeo y las ligeras diferencias en los resultados que producen estos comandos plantean pocos problemas. Sin embargo, si se desean más dígitos, el usuario puede controlar una variedad de opciones globales que dan forma a la forma en que **R** presenta los resultados con el comando de opciones que se mencionó en el Cap. 1 El argumento de los dígitos determina específicamente el número de dígitos presentados. Entonces, por ejemplo, podríamos escribir:

options(digits=9)  
  
summary(LL$re74)

Nuestra salida se vería así:

Min. 1er Qu. Mediana Media 3er Qu. Max.

0.000 0.000 823.822 3630.740 5211.790 39570.700

Por tanto, podemos ver que las discrepancias son una función del redondeo. Sin embargo, tenga en cuenta que los cambios con el comando de opciones se aplican a todas las salidas de la sesión. Por ejemplo, resulta que si volvemos a ejecutar el comando mean, la salida ahora mostraría incluso más dígitos de los que teníamos antes.

También podríamos obtener un resumen general de todas las variables en un conjunto de datos a la vez simplemente escribiendo el nombre del marco de datos solo en el comando de resumen :

summary(LL)

Esto informa las mismas estadísticas descriptivas que antes, pero para todas las variables a la vez. Si a alguna observación le falta un valor para una variable, este comando imprimirá el número de valores NA para la variable. Sin embargo, tenga en cuenta que no todas las cantidades informadas en esta tabla son significativas. Para las variables indicadoras como tratada , negra , casada , nodular , hispana , u74 y u75 , recuerde que las variables no son continuas. Básicamente, la media informa la proporción de encuestados que reciben un 1 en lugar de un 0, y el recuento de los valores faltantes es útil. Sin embargo, la otra información no es particularmente informativa.

### Tablas de frecuencia

Para las variables que se miden nominal u ordinalmente, el mejor resumen de información suele ser una tabla simple que muestra la frecuencia de cada valor. En **R**, el comando table nos informa de esto. Por ejemplo, nuestros datos incluyen un indicador simple codificado con 1 si el encuestado es afroamericano y 0 en caso contrario. Para obtener las frecuencias relativas de esta variable, escribimos:

table(LL$black)

Esto imprime la salida:

0 1

144 578

Por lo tanto, 144 encuestados no son afroamericanos y 578 encuestados son afroamericanos. Con un indicador nominal como este, la única medida válida de tendencia central es la moda , que es el valor más común que toma una variable. En este caso, el valor más frecuente es un 1, por lo que podríamos decir que la moda es afroamericana.

Como otro ejemplo, estos datos miden la educación en años. Medido de esta manera, podríamos pensar en esto como una variable continua. Sin embargo, el número de valores que adquiere esta variable es algo limitado, ya que nadie en la muestra tiene menos de tres o más de 16 años de educación. Por lo tanto, es posible que deseemos ver una tabla de frecuencias para esta variable también. Además, incluso si la media y la mediana son valores informativos, es posible que deseemos calcular la moda para saber cuál es el número más común de años de educación en estos datos. Para calcular la tabla de frecuencias y hacer que **R** devuelva automáticamente el modo, escribimos:

table(LL$education)  
  
which.max(table(LL$education))

La tabla que vemos es:

3 4 5 6 7 8 9 10 11 12 13 14 15 16

1 6 5 7 15 62 110 162 195 122 23 11 2 1

De un vistazo, pocos encuestados nunca fueron a la escuela secundaria y solo unos pocos tienen más que una educación secundaria. También podríamos escanear la tabla para observar que la moda es de 11 años de educación, lo que describe a 195 encuestados. Sin embargo, si tenemos muchas más categorías de las que tenemos para la educación , hacer esto será difícil. Por lo tanto, introducir nuestra tabla en el comando which.max devuelve qué etiqueta en la tabla tiene la frecuencia máxima . Nuestra impresión resultante es:

11

9

La primera línea, 11 , imprime la etiqueta de valor de la celda con la frecuencia más alta; este es nuestro modo. La segunda línea agrega el detalle adicional de que el valor de 11 es la novena celda de la tabla (un detalle que generalmente podemos ignorar).

Otra forma en que podríamos presentar nuestras frecuencias es en un diagrama de barras basado en la tabla anterior. Podríamos hacer esto con el siguiente código:

barplot(table(LL$education),xlab="Years of Education",  
  
 ylab="Frequency",cex.axis=.9,cex.names=.9,ylim=c(0,200))  
  
abline(h=0,col=’gray60’)  
  
box()

En la primera línea, especificamos que estamos dibujando un diagrama de barras de la tabla (educación LL $) . Observe que usamos cex.axis y cex.names para reducir el tamaño del texto en los ejes vertical y horizontal, respectivamente. Luego, agregamos una línea de base en 0 y dibujamos un cuadro alrededor de la figura completa. El resultado se muestra en la Fig.  4.2 . Con este gráfico, podemos detectar fácilmente que la barra más alta, nuestra moda, está en 11 años de educación. El gráfico también nos da una idea rápida de la dispersión de los otros valores. Abrir imagen en nueva ventanaFigura 4.2 Figura 4.2 Distribución del número de años de educación a partir de los datos de la demostración nacional de trabajo con apoyo

Como punto lateral, suponga que un analista no solo quiere una tabla de frecuencias, sino el porcentaje de valores en cada categoría. Esto podría lograrse simplemente dividiendo la tabla por la cantidad de casos y multiplicándola por 100. Entonces, para el porcentaje de encuestados que cae en cada categoría en la variable de educación, escribimos:

100\*table(LL$education)/sum(table(LL$education))

R luego imprimirá:

3 4 5 6

0.1385042 0.8310249 0.6925208 0.9695291

7 8 9 10

2.0775623 8.5872576 15.2354571 22.4376731

11 12 13 14

27.0083102 16.8975069 3.1855956 1.5235457

15 16

0.2770083 0.1385042

Este resultado ahora muestra el porcentaje de observaciones que pertenecen a cada categoría.

## Medidas de dispersión

Además de tener una idea del centro de nuestra variable, también nos gustaría saber qué tan dispersas están nuestras observaciones. Las medidas más comunes de esto son la varianza y la desviación estándar, aunque también discutiremos la desviación promedio promedio como una medida alternativa. Comenzando con la varianza de la muestra, nuestra fórmula para esta cantidad es:

En **R**, obtenemos esta cantidad con la función var . Para los ingresos en 1974, escribimos:

var(LL$re74)

Esto imprime el valor: [1] 38696328 . Por lo tanto, podemos escribir Var ( x ) = 38696328. Por supuesto, la varianza está en una métrica al cuadrado. Dado que es posible que no queramos pensar en el diferencial en términos de “38,7 millones de dólares al cuadrado,” también recurriremos a medidas alternativas de dispersión. Dicho esto, la varianza es una cantidad esencial que alimenta una variedad de otros cálculos de interés.

La desviación estándar es simplemente la raíz cuadrada de la varianza:

Esta simple transformación de la varianza tiene la buena propiedad de volver a poner nuestra medida de dispersión en la escala original. Podríamos tomar la raíz cuadrada de una varianza calculada o permitir que **R** haga todos los pasos del cálculo por nosotros:

sd(LL$re74)

En este caso, **R** imprime: [1] 6220,637 . Por lo tanto, s x  = 6220. 637. Cuando una variable tiene la forma de una distribución normal (que nuestra variable de ingreso no lo es), una aproximación útil es la regla 68-95-99.7. Esto significa que aproximadamente el 68% de nuestros datos se encuentran dentro de una desviación estándar de la media, el 95% dentro de dos desviaciones estándar y el 99,7% dentro de tres desviaciones estándar. Para el ingreso en 1974, una fuerte concentración de ingresos en $ 0 elimina esta regla, pero con muchas otras variables se mantendrá observacionalmente.

Una medida de dispersión muy diferente es la desviación absoluta mediana . Definimos esto como:

En este caso, usamos la mediana como nuestra medida de centralidad, en lugar de la media. Luego calculamos la diferencia absoluta entre cada observación y la mediana. Por último, calculamos la mediana de las desviaciones . Esto nos ofrece la sensación de una desviación típica de la mediana. En R, el comando se escribe:

mad(LL$re74)

Aquí, **R** devuelve un valor de 1221,398. Al igual que la desviación estándar, está en la escala de la variable original, en dólares. A diferencia de la desviación estándar, esta estadística resulta ser mucho menor en este caso. Una vez más, los valores extremos pueden aumentar las variaciones y las desviaciones estándar, al igual que pueden distorsionar una media. La desviación absoluta mediana, por el contrario, es menos sensible a los valores extremos.

### Cuantiles y percentiles

Como tema final, los cuantiles y percentiles nos permiten tener una idea de la distribución general de una variable. Los cuantiles son la ubicación relativa de los valores de datos en una lista ordenada, escalada [0, 1]. Para un valor q, el cuantil de ese valor sería el estadístico de orden x ( q ⋅  n ) . Los percentiles son lo mismo, escalados [0, 100], por lo que para un valor p, el percentil p sería X(p ⋅ n100). Por tanto, la mediana es el cuantil 0,5 y el percentil 50. Los casos especiales de cuantiles incluyen los cuartiles introducidos previamente (dividiendo los datos en cuatro grupos), quintiles (divididos en cinco grupos) y deciles (divididos en diez grupos).

En **R**, el cuantil de comando puede darnos cualquier cuantil que deseemos. De forma predeterminada, **R** imprime los cuantiles para q  ∈ {0. 00, 0. 25, 0. 50, 0. 75, 1. 00}. Sin embargo, tenemos la opción de especificar los cuantiles que queramos. La sintaxis es:

quantile(LL$re74)  
  
quantile(LL$re74, c(0,.1,.2,.3,.4,.5,.6,.7,.8,.9,1))

El primer comando imprime nuestros cuantiles predeterminados, aunque los informa con las etiquetas de percentiles reescaladas:

0% 25% 50% 75% 100%  
  
0.0000 0.0000 823.8215 5211.7946 39570.6797

Esencialmente, esta información repite la información de cuartiles que el resumen nos proporcionó anteriormente. En nuestra segunda línea de código, agregamos un vector de 11 cuantiles de interés para solicitar deciles, que nos dan los puntos de corte para cada 10% adicional de los datos. Este resultado es:

0% 10% 20% 30%  
  
0.0000 0.0000 0.0000 0.0000  
  
 40% 50% 60% 70%  
  
0.0000 823.8215 1837.2208 3343.5705  
  
 80% 90% 100%

6651.6747 10393.2177 39570.6797

Esto es revelador, ya que muestra que al menos el 40% de nuestros encuestados tenía un ingreso de $ 0 en 1974. Además, al pasar del percentil 90 al percentil 100 (o máximo), vemos un salto de $ 10,393 a $ 39,570, lo que sugiere que algunos Los valores particularmente extremos se encuentran en el 10% superior de nuestros datos. Por lo tanto, estos datos tienen un sesgo positivo sustancial, lo que explica por qué nuestra mediana calculada es tan diferente de la media.

En este capítulo, hemos cubierto los diversos medios por los cuales podemos calcular las medidas de centralidad y dispersión en **R**. También hemos discutido tablas de frecuencias y cuantiles. Junto con las técnicas de graficación del Cap. 3 , ahora tenemos una gran canasta de herramientas para evaluar y reportar los atributos de un conjunto de datos. En el próximo capítulo, pasaremos a hacer inferencias a partir de nuestros datos.

## Problemas de práctica

Considere de nuevo Peake y Eshbaugh-Soha (2008) análisis de la cobertura de la política de medicamentos, que se introdujo en los problemas de práctica del Cap. 3 . Recuerde que el archivo de datos separados por comas se llama drugCoverage.csv . Si aún no lo ha descargado, visite Dataverse (consulte la página vii) o el contenido en línea de este capítulo (consulte la página 53). Una vez más, las variables son: un índice de tiempo basado en caracteres que muestra el mes y el año ( año ), la cobertura de noticias sobre drogas ( drugmedia ), un indicador de un discurso sobre drogas que pronunció Ronald Reagan en septiembre de 1986 ( rwr86 ), un indicador de un discurso que pronunció George HW Bush en septiembre de 1989 ( ghwb89 ), el índice de aprobación del presidente ( aprobación ) y la tasa de desempleo ( desempleo ).

1. ¿Qué puede aprender simplemente aplicando el comando de resumen al conjunto de datos completo? ¿Qué salta más claramente de esta salida? ¿Hay algún valor perdido en estos datos?
2. Usando la función media , calcule lo siguiente:
   * ¿Cuál es la media del indicador del discurso de George HW Bush en 1989? Dado que esta variable solo toma valores de 0 y 1, ¿cómo interpretaría esta cantidad?
   * ¿Cuál es el nivel medio de aprobación presidencial? ¿Cómo interpretaría esta cantidad?
3. ¿Cuál es el nivel medio de cobertura de los medios de comunicación sobre temas relacionados con las drogas?
4. ¿Cuál es el rango intercuartílico de cobertura de los medios de comunicación sobre temas relacionados con las drogas?
5. Informe dos tablas de frecuencia:
   * En el primero, informe la frecuencia de los valores del indicador del discurso de Ronald Reagan de 1986.
   * En el segundo, informe la frecuencia de los valores de la tasa de desempleo en un mes determinado.
   * ¿Cuál es el valor modal del desempleo? ¿En qué porcentaje de meses ocurre la moda?
6. ¿Cuáles son la varianza, la desviación estándar y la desviación absoluta mediana para la cobertura de noticias sobre drogas?
7. ¿Cuáles son los percentiles 10 y 90 de aprobación presidencial en este período de tiempo 1977-1992?

## Referencias

* LaLonde RJ (1986) Evaluación de las evaluaciones econométricas de programas de formación con datos experimentales. Am Econ Rev 76 (4): 604–620
* Peake JS, Eshbaugh-Soha M (2008) El impacto en el establecimiento de la agenda de los principales discursos televisivos presidenciales. Polit Commun 25: 113-137

# 5 Inferencias básicas y asociación bivariante

Palabras clave: - Estado de Empleo - Hipótesis alternativa - Estimación de la muestra - Estadística inferencial - Intervalo de confianza porcentual

En este capítulo, comenzamos a utilizar la estadística inferencial y la estadística bivariada . En el Cap. 4 , nos contentamos simplemente con caracterizar las propiedades de una sola variable en la muestra en cuestión. Sin embargo, normalmente en Ciencias Políticas, nuestra motivación como investigadores será discutir si una afirmación puede generalizarse. Por lo tanto, las estadísticas inferenciales están diseñadas para hacer inferencias sobre una población más amplia. Además, con frecuencia queremos medir el nivel de asociación entre variables, y las estadísticas bivariadas sirven como medidas del grado en que dos variables están asociadas entre sí.

En capítulos posteriores, los modelos de regresión lineal, los modelos lineales generalizados, los modelos de series de tiempo y otros modelos que estimamos ofrecen la oportunidad de hacer una inferencia sobre una población más amplia. También nos permiten evaluar relaciones bivariadas o multivariadas entre variables. Por ahora, nos enfocamos en un puñado de estadísticos inferenciales y bivariados trampolín: pruebas sobre medias, asociaciones entre dos variables categóricas (a través de tabulaciones cruzadas) y correlaciones entre dos variables continuas.

En este capítulo, los datos de nuestro ejemplo de trabajo serán los mismos datos de LaLonde (1986) análisis de la Manifestación Nacional de Trabajo Apoyado. La información sobre las características de estos datos se puede revisar al comienzo del Cap. 4 . En este caso, cada miembro de nuestra muestra es alguien que estuvo desempleado de larga duración. Por lo tanto, al hacer inferencias, no sería justo intentar llegar a una conclusión sobre toda la población de los EE. UU. A mediados de la década de 1970 porque estos datos no componen una muestra de esa población. A los efectos de nuestro ejemplo práctico, intentaremos sacar conclusiones sobre la población de desempleados de larga duración en los EE. UU. Más información sobre cómo se extrajo esta muestra está disponible en LaLonde ( 1986), y se insta al lector a leer sobre muestreo aleatorio si desea obtener más información sobre la teoría de la inferencia estadística.

## Pruebas de significancia para medias

Antes de calcular cualquier estadística inferencial, debemos cargar los datos de LaLonde una vez más. Los usuarios que ya han instalado la biblioteca cem pueden simplemente escribir:

biblioteca (cem)

datos (LL)

Los usuarios que no instalaron cem en el Cap. 4 deberá escribir install.packages (“cem”) antes de que las dos líneas de código anteriores funcionen correctamente. Una vez que estos datos se cargan en la memoria, nuevamente con el nombre LL , podemos pasar al análisis aplicado 1 .

Comenzamos probando hipótesis sobre la media de una población (o poblaciones múltiples). Primero consideramos el caso en el que queremos probar si la media de alguna población de interés difiere de algún valor de interés. Para realizar esta prueba de significancia, necesitamos: (1) nuestra estimación de la media muestral, (2) el error estándar de nuestra estimación media y (3) una hipótesis nula y alternativa. La media muestral se definió anteriormente en la ecuación. ( 4.1 ), y el error estándar de nuestra estimación es simplemente la desviación estándar de la variable [definida en la Ec. ( 4.5 )] dividido por la raíz cuadrada del tamaño de nuestra muestra, sX/norte–√ .

Al definir nuestras hipótesis nula y alternativa, definimos la hipótesis nula en función de algún valor de interés que nos gustaría descartar como valor posible del parámetro de población. Por tanto, si decimos: H0:μ =μ0 Esto significa que nuestra hipótesis nula ( H 0 ) es que la media poblacional ( μ ) es igual a algún valor numérico que establezcamos ( μ 0 ). Nuestra hipótesis de investigación es la hipótesis alternativa que nos gustaría rechazar esta nula a favor. Tenemos tres opciones para posibles hipótesis de investigación: HA:HA:HA:μ >μ0μ <μ0μ ≠μ0 Las dos primeras se denominan pruebas de una cola e indican que creemos que la media de la población debería ser, respectivamente, mayor o menor que el valor propuesto μ 0 . La mayoría de las hipótesis de investigación deben considerarse como una de las pruebas de una cola, aunque ocasionalmente el analista no tiene una expectativa fuerte sobre si la media debe ser mayor o menor. La tercera alternativa enumerada define la prueba de dos colas, que pregunta si la media es simplemente diferente (o no igual) del valor μ 0 . Una vez que hemos formulado nuestra hipótesis, calculamos una relación t como nuestro estadístico de prueba para la hipótesis. Nuestro estadístico de prueba incluye la media muestral, el error estándar y la media poblacional definida por la hipótesis nula ( μ 0 ). Esta fórmula es: t =X¯-μ0S E (X¯|H0)=X¯-μ0sX/norte–√ (5,1) Esta se distribuye t de Student con n - 1 grados de libertad bajo el valor nulo (y asintóticamente normal). 2 Una vez que tenemos esta estadística de prueba, calculamos nuestro valor p de la siguiente manera: p - v a l u e =⎧⎩⎨⎪⎪PAG(t∗≤ t |H0)PAG(t∗≥ t |H0)PAG( |t∗-μ0| ≥ | t -μ0| |H0)HA: μ <μ0HA: μ >μ0HA: μ ≠μ0 En este caso, suponga que t ∗ es el valor real de nuestra estadística que calculamos. La acción típica en este caso es tener un nivel de confianza predefinido y decidir si rechazar la hipótesis nula o no en función de si el valor p indica que el rechazo se puede realizar con ese nivel de confianza. Por ejemplo, si un analista estuviera dispuesto a rechazar una hipótesis nula si pudiera hacerlo con un 90% de confianza, entonces si p  <0,10, rechazaría la hipótesis nula y concluiría que la hipótesis de investigación es correcta. Muchos usuarios también proceden a informar el valor p para que los lectores puedan sacar conclusiones sobre la importancia por sí mismos. R hace que todos estos cálculos sean muy sencillos, haciendo todo esto en una sola línea de código de usuario. Supongamos que tuviéramos la hipótesis de que, en 1974, la población de estadounidenses desempleados a largo plazo tenía un ingreso inferior a $ 6.059, una estimación del gobierno del ingreso medio para la población general de estadounidenses. En este caso, nuestra hipótesis es: H0:HA:μ = 6059μ < 6059 Esta es una prueba de una sola cola porque ni siquiera pensamos en la idea de que los desempleados de larga duración puedan tener un ingreso promedio más alto que la población en general. Más bien, simplemente preguntamos si la media de nuestra población de interés es perceptiblemente inferior a 6.059 dólares o no. Para probar esta hipótesis en R , escribimos: t.test (LL $ re74, mu = 6059, alternativa = “menos”)

El primer argumento de la prueba t enumera nuestra variable de interés, LL $ re74 , para la cual R calcula automáticamente la media muestral y el error estándar. En segundo lugar, el argumento mu = 6059 enumera el valor de interés de nuestra hipótesis nula. Asegúrese de incluir este argumento: si lo olvida, el comando aún se ejecutará asumiendo que quiere mu = 0 , lo cual es una tontería en este caso. Finalmente, especificamos nuestra hipótesis alternativa como “menos” . Esto significa que creemos que la media de la población es menor que la cantidad nula presentada. El resultado de este comando se imprime como:

Prueba t para una muestra

datos: LL $ re74

t = -10.4889, gl = 721, valor de p <2.2e-16

hipótesis alternativa: la media verdadera es menor que 6059

Intervalo de confianza del 95 por ciento:

-Inf 4012.025

estimaciones de muestra:

media de x

3630.738

Esto presenta una larga lista de información: al final, informa la media muestral de 3630,738. Anteriormente, nos muestra que el valor de nuestra relación t es - 10. 4889, junto con el hecho de que nuestra distribución t tiene 721 grados de libertad. En cuanto al valor p , cuando R imprime el valor p <2.2e-16 , esto significa que p es tan minúsculo que es más pequeño que el nivel de precisión decimal de R , mucho menos cualquier umbral de significación común. Por lo tanto, podemos rechazar la hipótesis nula y concluir que los estadounidenses desempleados de larga duración tenían un ingreso significativamente más bajo que los $ 6.059 en 1974.

5.1.1 Prueba de diferencia de medias de dos muestras, muestras independientes Como alternativa al uso de una muestra para hacer una inferencia sobre la media de la población relevante, podemos tener dos muestras y querer probar si las medias de las dos poblaciones son iguales. En este caso, si llamamos al conjunto de observaciones de una muestra xy a las observaciones de la segunda muestra y , entonces formularíamos nuestra hipótesis nula como: H0:μX=μy Nuevamente, emparejaremos esto con una de las tres hipótesis alternativas: HA:HA:HA:μXμyμX≠μy Una vez más, las dos primeras hipótesis alternativas posibles son pruebas de una cola en las que tenemos una expectativa clara sobre qué media de la población debería ser mayor. La tercera alternativa posible simplemente evalúa si las medias son diferentes. Al construir nuestro estadístico de prueba a partir de esta hipótesis nula, confiamos en el hecho de que H 0 también implica μ x - μ y  = 0. Usando este hecho, construimos nuestra relación t como: t =(X¯-y¯) - (μX-μy)S E (X¯-y¯|H0) (5,2) La última pregunta es cómo calculamos el error estándar. Nuestro cálculo depende de si estamos dispuestos a asumir que la varianza es la misma en cada población. Bajo el supuesto de varianza desigual, calculamos el error estándar como: S E (X¯-y¯|H0) =s2XnorteX+s2ynortey——–√ (5,3) Bajo el supuesto de igual varianza, tenemos: S E (X¯-y¯|H0) =(norteX- 1 )s2X+ (nortey- 1 )s2ynorteX+nortey- 2——————-√1norteX+1nortey——–√ (5,4) Como ejemplo, podemos realizar una prueba con la última observación de ingresos en la Demostración Nacional de Trabajo Apoyado, que se midió en 1978. Supongamos que nuestra hipótesis es que el ingreso en 1978 fue mayor entre los individuos que recibieron el tratamiento de participar en el programa ( y ) que entre los que fueron observaciones de control y no llegaron a participar en el programa ( x ). Nuestra hipótesis en este caso es: H0:HA:μX=μyμX<μy Una vez más, esta es una prueba de una cola porque no estamos considerando la idea de que el tratamiento podría haber reducido los ingresos a largo plazo. Más bien, el tratamiento aumentó los ingresos en relación con las observaciones de control o no tuvo un efecto perceptible. R nos permite realizar esta prueba t de dos muestras utilizando cualquiera de los supuestos. Los comandos para variaciones desiguales e iguales, respectivamente, son: t.test (re78 ~ tratado, datos = LL, alternativa = “menos,” var.equal = F)

t.test (re78 ~ tratado, datos = LL, alternativa = “menos,” var.equal = T)

El primer argumento, tratado nuevamente , está en notación funcional e indica que los ingresos en 1978 se están separando en función de los valores del indicador de tratamiento. La opción de datos nos permite nombrar el conjunto de datos para que no tengamos que llamarlo para cada variable. Cuando establecemos alternativa = “menos” , estamos declarando que nuestra hipótesis alternativa significa que el ingreso promedio para el valor más bajo de los tratados (grupo 0, el control) debe ser menor que el promedio para el valor más alto de los tratados (grupo 1, el grupo tratado). La única diferencia entre los comandos es que el primero establece var.equal = F de modo que las varianzas se asumen desiguales, y el segundo establece var.equal = T de modo que las varianzas se asumen iguales.

Los resultados se imprimen de la siguiente manera. Para el supuesto de varianzas desiguales, vemos:

Prueba t de dos muestras de Welch

datos: re78 por tratado

t = -1.8154, gl = 557.062, valor p = 0.035

hipótesis alternativa: la verdadera diferencia de medias es menor

de 0

Intervalo de confianza del 95 por ciento:

-Inf -81.94117

estimaciones de muestra:

media en el grupo 0 media en el grupo 1

5090.048 5976.352

Mientras tanto, para el supuesto de igual varianza, vemos:

Prueba t de dos muestras

datos: re78 por tratado

t = -1,8774, gl = 720, valor p = 0,03043

hipótesis alternativa: la verdadera diferencia de medias es menor

de 0

Intervalo de confianza del 95 por ciento:

-Inf -108.7906

estimaciones de muestra:

media en el grupo 0 media en el grupo 1

5090.048 5976.352

Los resultados son bastante similares, como era de esperar. Ambos informan las mismas estimaciones de las medias para el grupo de control (5090.048) y el grupo tratado (5976.352). Sin embargo, debido a que el error estándar se calcula de manera diferente, obtenemos valores t ligeramente diferentes en cada caso (y grados de libertad calculados de manera diferente para las varianzas desiguales). La p-valor es ligeramente mayor cuando se asumen varianzas desiguales, pero en este caso cualquiera de las opciones arroja una conclusión similar. Por lo tanto, en cualquier caso, rechazamos la hipótesis nula al nivel de confianza del 95% y concluimos que para el grupo tratado el ingreso fue mayor en 1978 que para el grupo de control. Cabe señalar que una limitación de una prueba como esta es que no hemos controlado ninguna de las otras variables que se sabe que influyen en los ingresos, y la asignación de tratamiento no fue aleatoria en este caso. Capítulos 6 - 8 ofrecen varios ejemplos de métodos diseñados para controlar estadísticamente para otros predictores. Específicamente, en la Sect. 8.3 volvemos a visitar este ejemplo exacto con una técnica más avanzada.

5.1.2 Comparación de medias con muestras dependientes Un tercer estadístico de prueba relacionado con la media que quizás deseemos calcular es una diferencia de medias con una muestra dependiente (por ejemplo, comparando muestras emparejadas). En otras palabras, suponga que tenemos una situación en la que cada observación de la muestra 1 coincide con una observación de la muestra 2. Esto podría significar que estamos estudiando a la misma persona antes y después de un evento, una persona que realiza la misma tarea con diferentes tratamientos, utilizando un estudio de gemelos, o el uso de métodos de emparejamiento para emparejar las observaciones tratadas con las observaciones de control. En este caso, ya no deberíamos tratar cada muestra como independiente, sino calcular las diferencias para cada emparejamiento y analizar las diferencias. Una manera fácil de pensar en esto sería crear una nueva variable, w i  =  x i - y i donde x y y están adaptadas para cada caso i . En este caso, nuestra hipótesis nula es H 0 :  μ w  = 0, y nuestra hipótesis alternativa puede ser cualquiera de las tres opciones: HA:HA:HA:μw< 0μw> 0μw≠ 0 El estadístico de prueba en este caso viene dado por la Ec. ( 5.5 ), calculado para la nueva variable w . t =w¯- 0S E (w¯|H0)=w¯sw/norte–√ (5,5) Como puede verse, este es efectivamente el mismo estadístico de prueba que en la Ec. ( 5.1 ) con w como variable de interés y 0 como valor nulo. El usuario técnicamente podría crear la variable w por sí mismo y luego simplemente aplicar el código para una prueba de significancia de muestra única para una media. Sin embargo, más rápidamente, este procedimiento podría automatizarse insertando dos variables separadas para las observaciones vinculadas en el comando t -test. Supongamos, por ejemplo, que quisiéramos saber si nuestras observaciones de control vieron un aumento en sus ingresos de 1974 a 1978. Es posible que los salarios no aumenten durante este tiempo porque estos números se registran en términos reales. Sin embargo, si los salarios aumentaron, entonces observar cómo cambiaron para el grupo de control puede servir como una buena base para comparar el cambio en los salarios del grupo tratado en el mismo período de tiempo. Para realizar esta prueba t de muestra pareada para nuestras observaciones de control, escribimos:

LL.0 <-subconjunto (LL, tratado == 0)

t.test (LL.0 $ re74, LL.0 $ re78, emparejado = T, alternativa = “menos”)

En la primera línea creamos un subconjunto solo de nuestras observaciones de control. En la segunda línea, nuestro primer argumento es la medida del ingreso en 1974, y el segundo es el ingreso en 1978. En tercer lugar, especificamos la opción paired = T : esto es crítico , de lo contrario R asumirá que cada variable forma una muestra independiente, pero en nuestro caso se trata de una muestra pareada en la que cada individuo ha sido observado dos veces. (Con este fin, al escribir paired = F en su lugar, esto nos da la sintaxis para una prueba t de dos muestras si nuestras muestras separadas están en columnas de datos diferentes). Finalmente, alternativa = “menos” significa que esperamos que la media de la primera observación, en 1974, sea menor que la media de la segunda, en 1978. Nuestros resultados son:

Prueba t pareada

datos: LL.0 $ re74 y LL.0 $ re78

t = -3.8458, gl = 424, valor p = 6.93e-05

hipótesis alternativa: la verdadera diferencia de medias es

menos de 0

Intervalo de confianza del 95 por ciento:

-Inf -809.946

estimaciones de muestra:

media de las diferencias

-1417.563

Este resultado nos dice que las ganancias fueron en promedio $ 1,417.56 más bajas en 1974 que en 1978. Nuestra relación t es t  = −3. 8458, y el valor p correspondiente es p  = 0. 00007. Por lo tanto, en cualquier umbral de confianza común, podemos rechazar la hipótesis nula y concluir que los ingresos eran más altos en 1978 que en 1974 entre los desempleados de larga duración que no recibieron el tratamiento .

Solo como comparación final, podríamos calcular el mismo tipo de prueba en el grupo tratado de la siguiente manera:

LL.1 <-subconjunto (LL, tratado == 1)

t.test (LL.1 $ re74, LL.1 $ re78, emparejado = T, alternativa = “menos”)

Los resultados son algo similares:

Prueba t pareada

datos: LL.1 $ re74 y LL.1 $ re78

t = -4.7241, gl = 296, valor p = 1.788e-06

hipótesis alternativa: la verdadera diferencia de medias es

menos de 0

Intervalo de confianza del 95 por ciento:

-Inf -1565.224

estimaciones de muestra:

media de las diferencias

-2405,353

Observamos un crecimiento mayor de $ 2,405.35 entre las observaciones tratadas, y este resultado también es discernible estadísticamente. Esta mayor tasa de crecimiento es alentadora para el potencial de ganancias a largo plazo del programa. Para obtener puntos de bonificación, se anima al lector a buscar técnicas de diferencias en diferencias y considerar cómo podrían aplicarse a estudios con un diseño como este.

5.2 Tabulaciones cruzadas En situaciones en las que queremos analizar la asociación entre dos variables nominales u ordinales, una tabulación cruzada suele ser una buena herramienta para la inferencia. Una tabulación cruzada prueba la hipótesis de que una variable categórica independiente afecta la distribución condicional de una variable categórica dependiente. El investigador pregunta: ¿Serán notablemente más o menos frecuentes ciertos valores de la variable dependiente al pasar de una categoría de una variable independiente a otra? Al evaluar este efecto, el analista siempre debe desglosar los porcentajes relativos de las categorías de la variable dependiente dentro de las categorías de la variable independiente. Mucha gente comete un error al desglosar los porcentajes dentro de las categorías de la variable dependiente; tal error impide que un investigador evalúe sustancialmente la hipótesis establecida de que la variable independiente causa la variable dependiente y no al revés. Los resultados de una tabulación cruzada comparan sustancialmente los porcentajes del mismo valor de la variable dependiente en todas las categorías de la variable independiente.

Algunos errores comunes que se deben evitar: Primero, una vez más, evite desglosar los porcentajes por la variable dependiente. En segundo lugar, evite comparar el mayor porcentaje en cada categoría de una variable independiente. La hipótesis establece que la frecuencia de la variable dependiente variará según el valor de la variable independiente; no discute qué valor de la variable dependiente será más frecuente. Por último, evite hacer inferencias basadas en la magnitud pura de porcentajes; la tarea del investigador es observar las diferencias en la distribución. Por ejemplo, si la elección del voto es la variable dependiente y el 66% de los republicanos apoya al candidato demócrata, mientras que el 94% de los demócratas apoya al candidato demócrata, el investigador no debe centrarse en el apoyo de la mayoría de ambos partidos. En lugar de,

Considere dos ejemplos del conjunto de datos de LaLonde. Primero, podemos simplemente preguntarnos si estar desempleado en 1974 ( u74 ) sirvió como un buen predictor de estar desempleado en 1975 ( u75 ). Tendríamos que pensar que el estado laboral anterior de un individuo determina el estado laboral actual. Para construir una tabulación cruzada en R , necesitamos instalar el paquete gmodels y luego cargar la biblioteca. Una vez hecho esto, podemos usar la función CrossTable :

install.packages (’’ gmodels ’’)

biblioteca (gmodels)

Tabla cruzada (y = LL $ u75, x = LL $ u74, prop.c = F, prop.t = F,

prop.chisq = F, chisq = T, formato = "SPSS")

En este código, y especifica la variable de columna y x especifica la variable de fila. Esto significa que nuestra variable dependiente forma las columnas y la independiente forma las filas. Porque queremos que la distribución condicional de la variable dependiente para cada valor dado de la variable independiente, el opciones prop.c , prop.t y prop.chisq se ajustan a FALSO (en referencia a apuntalar Ortion de la c OLUMNA, t otal muestra y contribución al chisquare estadística). Esto significa que cada celda solo contiene la frecuencia bruta y el porcentaje de fila, que corresponde a la distribución condicionada a la variable independiente. La opción chisq = T informa la prueba Chi-cuadrado ( χ 2 ) de Pearson . Bajo esta prueba, la hipótesis nula es que las dos variables son independientes entre sí. La hipótesis alternativa es que conocer el valor de una variable cambia la distribución esperada de la otra. 3 Al establecer la opción de formato en SPSS , en lugar de SAS , se nos presentan porcentajes en nuestras celdas, en lugar de proporciones.

Los resultados de este comando se imprimen a continuación:

Contenido de la celda

————————- |

Contar |

Porcentaje de fila |

————————- |

Total de observaciones en la tabla: 722

| LL $ u75  
  
 LL $ u74 | 0 | 1 | Total de filas |

————- | ———– | ———– | ———– |

0 | 386 | 9 | 395 |  
  
 | 97,722% | 2,278% | 54,709% |

————- | ———– | ———– | ———– |

1 | 47 | 280 | 327 |  
  
 | 14,373% | 85,627% | 45,291% |

————- | ———– | ———– | ———– |

Total de la columna | 433 | 289 | 722 |

————- | ———– | ———– | ———– |

Estadísticas para todos los factores de la tabla

Prueba de chi-cuadrado de Pearson

Chi ^ 2 = 517.7155 gl = 1 p = 1.329138e-114

Prueba de chi-cuadrado de Pearson con continuidad de Yates

corrección

Chi ^ 2 = 514.2493 gl = 1 p = 7.545799e-114

Frecuencia mínima esperada: 130.8906

Como podemos ver, entre los que estaban empleados en 1974 ( u74 = 0), el 97,7% estaban empleados en 1975. Entre los que estaban desempleados en 1974 ( u75 = 1), el 14,4% estaban empleados en 1975. 4 Esto corresponde a un 83,3 puntos porcentuales de diferencia entre las categorías. Este vasto efecto indica que la situación laboral en 1 año, de hecho, engendra una situación laboral en el año siguiente. Además, nuestra estadística de prueba es χ21 d f= 517,7155con un minúsculo valor p correspondiente . Por tanto, rechazamos la hipótesis nula de que la situación laboral en 1974 es independiente de la situación laboral en 1975 y concluimos que la situación laboral en 1974 condiciona la distribución de la situación laboral en 1975.

Como pregunta más interesante, podríamos preguntarnos si recibir el tratamiento de la Demostración Nacional de Trabajo con Apoyo da forma a la situación laboral en 1975. Probaríamos esta hipótesis con el código:

Tabla cruzada (y = LL $ u75, x = LL $ tratado, prop.c = F, prop.t = F,

prop.chisq = F, chisq = T, formato = "SPSS")

El resultado de este comando es:

Contenido de la celda

————————- |

Contar |

Porcentaje de fila |

————————- |

Total de observaciones en la tabla: 722

| LL $ u75

LL $ tratado | 0 | 1 | Total de filas |

————- | ———– | ———– | ———– |

0 | 247 | 178 | 425 |  
  
 | 58,118% | 41,882% | 58,864% |

————- | ———– | ———– | ———– |

1 | 186 | 111 | 297 |  
  
 | 62,626% | 37,374% | 41,136% |

————- | ———– | ———– | ———– |

Total de la columna | 433 | 289 | 722 |

————- | ———– | ———– | ———– |

Estadísticas para todos los factores de la tabla

Prueba de chi-cuadrado de Pearson

Chi ^ 2 = 1.480414 gl = 1 p = 0.2237097

Prueba de chi-cuadrado de Pearson con continuidad de Yates

corrección

Chi ^ 2 = 1.298555 gl = 1 p = 0.2544773

Frecuencia mínima esperada: 118,8823

Sustancialmente, los efectos están en la dirección esperada. Entre las observaciones de control ( tratadas = 0), el 58,1% se empleó en 1975. Entre las observaciones tratadas ( tratadas = 1), el 62,6% se empleó en 1975. Por lo tanto, vemos un aumento de 4,5 puntos porcentuales en el empleo entre las observaciones tratadas sobre las observaciones de control. Sin embargo, nuestra estadística de prueba es χ21 d f= 1.4804. El valor p correspondiente es p  = 0. 2237. Por lo tanto, si establecemos nuestro umbral de confianza en 90% o algo más alto, no rechazaríamos la hipótesis nula y concluiríamos que no había una relación discernible entre el tratamiento y la situación laboral.

5.3 Coeficientes de correlación Como un adelanto al siguiente capítulo, concluimos nuestra mirada a las estadísticas de dos variables, mostrando cómo calcular un coeficiente de correlación de R . Los coeficientes de correlación se calculan como una medida de asociación entre dos variables continuas. Nos centramos específicamente en la r de Pearson , el coeficiente de correlación para una relación lineal. Este valor muestra qué tan bien la variable independiente predice linealmente la variable dependiente. Esta medida variará entre -1 y 1. Un coeficiente de correlación de 0 sugiere la ausencia de cualquier relación lineal entre las dos variables. (Aunque, lo que es más importante, una relación no lineal también puede producir r = 0 y algunas conclusiones erróneas.) Un valor de 1 implicaría una relación positiva perfecta, y un valor de - 1 indicaría una relación negativa perfecta. El cuadrado de la r de Pearson ( r 2 ) calcula la cantidad de varianza explicada por el predictor.

La fórmula para un coeficiente de correlación de Pearson es esencialmente la covarianza de dos variables, x y y , dividido por la desviación estándar de cada variable: r =∑nortei = 1(XI-X¯) (yI-y¯)∑nortei = 1(XI-X¯)———–√∑nortei = 1(yI-y¯)———–√ (5,6) Dentro de R , esta cantidad se calcula con el comando cor . 5 Supongamos que quisiéramos evaluar si el número de años de educación sirvió como un buen predictor de nuestra primera medida de ingresos, en 1974. Podríamos escribir:

cor (LL $ educación, LL $ re74)

cor (LL $ educación, LL $ re74) ^ 2

La primera línea calcula el propio coeficiente de correlación real. R devuelve una impresión de: [1] 0.08916458 . Por lo tanto, nuestro coeficiente de correlación es r  = 0. 0892. La segunda línea recalcula la correlación y eleva al cuadrado el resultado para todos a la vez. Esto nos dice que r 2  = 0. 0080. La implicación de este hallazgo es que al conocer el número de años de educación de un encuestado, podríamos explicar el 0,8% de la varianza en los ingresos de 1974. A primera vista, esto parece algo débil, pero como consejo general, siempre mida r 2 (o múltiples R 2, en el capítulo siguiente) valores comparándolos con otros hallazgos en la misma área. Algunos tipos de modelos explicarán habitualmente el 90% de la varianza, mientras que otros lo harán bien para explicar el 5% de la varianza.

Como ejemplo final, podemos considerar la idea de que los ingresos engendran ingresos. Considere qué tan bien se correlacionan los ingresos en 1975 con los ingresos en 1978. Calculamos esto escribiendo:

cor (LL $ re75, LL $ re78)

cor (LL $ re75, LL $ re78) ^ 2

La primera línea devuelve el coeficiente de correlación entre estas dos variables, imprimiendo: [1] 0.1548982 . Nuestra estimación de r  = 0. 1549 indica que los valores altos de ingresos en 1975 corresponden generalmente a valores altos de ingresos en 1978. En este caso, la segunda línea devuelve r 2  = 0. 0240. Esto significa que podemos explicar el 2.4% de la varianza del ingreso en 1978 al saber lo que alguien ganó en 1975.

Recuerde que las herramientas gráficas del Cap. 3 puede ayudarnos a comprender nuestros datos, incluidos los resultados que cuantificamos, como los coeficientes de correlación. Si nos preguntamos por qué los ingresos anteriores no predicen mejor los ingresos posteriores, podríamos dibujar un diagrama de dispersión de la siguiente manera:

plot (x = LL $ re75, y = LL $ re78, xlab = “Ingresos de 1975,” ylab = “Ingresos de 1978,”

asp = 1, xlim = c (0,60000), ylim = c (0,60000), pch = ".")

Tenga en cuenta que se ha utilizado el asp = 1 opción para establecer el asp relación ect de los dos ejes en 1. Esto garantiza que la escala de los dos ejes se mantiene para ser el mismo que es apropiado, ya que se miden ambas variables en la figura en dólares ajustados a la inflación. La salida se muestra en la Fig.  5.1 . Como puede verse, muchas de las observaciones se agrupan en cero en uno o ambos años, por lo que existe un grado limitado en el que una relación lineal caracteriza estos datos. Abrir imagen en nueva ventanaFigura 5.1 Figura 5.1 Diagrama de dispersión de los ingresos en 1975 y 1978 a partir de los datos de la demostración nacional de trabajo con apoyo

Ahora tenemos varias inferencias básicas en la mano: pruebas t sobre medias y pruebas χ 2 para tabulaciones cruzadas. La diferencia en las pruebas de medias, las tabulaciones cruzadas y los coeficientes de correlación también nos han dado un buen sentido para evaluar las relaciones bivariadas. En el próximo capítulo, pasaremos a la estadística multivariante, específicamente utilizando métodos de regresión lineal. Esto se basará en las técnicas lineales que utilizan los coeficientes de correlación y nos permitirá introducir el concepto de control estadístico.

5.4 Problemas de práctica Cargue la biblioteca extranjera y descargue la de Alvarez et al. (2013), que se guardan en el archivo con formato Stata alpl2013.dta . Este archivo está disponible en el Dataverse mencionado en la página vii o en el contenido del capítulo mencionado en la página 63. Estos datos provienen de un experimento de campo en Salta, Argentina, en el que algunos votantes emitieron sus votos a través del voto electrónico y otros votaron en el entorno tradicional. Las variables son: un indicador de si el votante utilizó el voto electrónico o el voto tradicional ( EV ), grupo de edad ( age\_group ), educación ( educ ), trabajador de cuello blanco ( white\_collar ), no un trabajador de tiempo completo ( not\_full\_time ), hombre ( masculino ), una variable de conteo para el número de seis posibles dispositivos tecnológicos utilizados ( tecnología), una escala ordinal para el conocimiento político ( pol\_info ), un vector de caracteres que nombra el lugar de votación ( polling\_place ), si el encuestado piensa que los trabajadores electorales están calificados ( able\_auth ), si el votante evaluó la experiencia de votación positivamente ( eval\_voting ), si el votante evaluó la velocidad de la votación como rápida ( speed ), si el votante está seguro de que su voto está siendo contado ( sure\_counted ), si el votante pensó que votar era fácil ( easy\_voting ), si el votante confía en el secreto de la boleta ( conf\_secret ), si el votante piensa que las elecciones de Salta están limpias ( how\_clean), si el votante piensa que el voto electrónico debería reemplazar el voto tradicional ( accept\_evoting ), y si el votante prefiere seleccionar candidatos de diferentes partidos electrónicamente ( eselect\_cand ). 1. Considere la cantidad de dispositivos tecnológicos. Pruebe la hipótesis de que el votante salteño promedio ha utilizado más de tres de estos seis dispositivos. (Formalmente: H 0 :  μ  = 3;  H A :  μ  > 3.)

1. Realice dos pruebas de diferencia de medias de muestras independientes: un. ¿Existe alguna diferencia entre hombres y mujeres en la cantidad de dispositivos tecnológicos que han utilizado?
2. ¿Existe alguna diferencia en la forma en que los votantes ven positivamente la experiencia de votar ( eval\_voting ) en función de si utilizaron el voto electrónico o el voto tradicional ( EV )?
3. Construya dos tabulaciones cruzadas: un. Construya una tabulación cruzada donde la variable dependiente es qué tan positivamente ven los votantes la experiencia de votación ( eval\_voting ) y la variable independiente es si usaron el voto electrónico o el voto tradicional ( EV ). ¿Depende la distribución de la evaluación del voto de si el votante utilizó el voto electrónico? Esta tabulación cruzada abordó la misma pregunta que se plantea en el # 2.b. ¿Qué enfoque es más apropiado aquí?
4. Construya una tabulación cruzada donde la variable dependiente es qué tan positivamente ven los votantes la experiencia de votar ( eval\_voting ) y la variable independiente es la escala ordinal de conocimiento político ( pol\_info ). ¿Cambia la distribución de la evaluación del voto con el nivel de conocimiento político del votante?
5. Considere la correlación entre el nivel de educación ( educ ) y el conocimiento político ( pol\_info ): un. Calcule la r de Pearson entre estas dos variables.
6. Muchos argumentan que, con dos variables ordinales, una medida de correlación más apropiada es la ρ de Spearman , que es una correlación de rango. Calcule ρ y contraste los resultados de r .

Notas al pie 1 . Como antes, estos datos también están disponibles en formato separado por comas en el archivo llamado LL.csv . Este archivo de datos se puede descargar del Dataverse en la página vii o del enlace de contenido del capítulo en la página 63.

2 . Este estadístico tiene una distribución t porque la media muestral tiene una distribución muestral distribuida normalmente y el error estándar muestral tiene una distribución muestral χ 2 con n - 1 grados de libertad. La razón de estas dos distribuciones produce una distribución t .

3 . Tenga en cuenta que esta es una prueba simétrica de asociación. La prueba en sí no tiene noción de cuál es la variable dependiente o independiente.

4 . Para obtener niveles más significativos que 0 y 1 en este caso, necesitaríamos crear copias de las variables u74 y u75 que registraron cada valor como texto (por ejemplo, “Desempleado” y “Empleado”). El comando recodificar de la biblioteca de automóviles ofrece una forma sencilla de hacer esto, si lo desea.

5 . El comando cor también proporciona una opción de método para la cual los argumentos disponibles son pearson , kendall (que calcula τ de Kendall , una correlación de rango) y spearman (que calcula ρ de Spearman , otra correlación de rango). Se anima a los usuarios a leer sobre los métodos alternativos antes de utilizarlos. Aquí, nos centramos en el método de Pearson predeterminado.

Material suplementario 318886\_1\_En\_5\_MOESM1\_ESM.zip (1 mb) Dataverse (2,154 KB) Referencias Alvarez RM, Levin I, Pomares J, Leiras M (2013) Votar de forma segura y sencilla: el impacto del voto electrónico en la percepción ciudadana. Polit Sci Res Methods 1 (1): 117-137 CrossRefGoogle Académico LaLonde RJ (1986) Evaluación de las evaluaciones econométricas de programas de formación con datos experimentales. Am Econ Rev 76 (4): 604–620 Google Académico

# 6 Modelos lineales y diagnósticos de regresión

Palabras clave: - Cuadrados mínimos ordinarios - Cuadrado ponderado - Estimación de mínimos cuadrados ordinarios - Coeficiente de regresión parcial - Mejor estimador lineal imparcial

El modelo de regresión lineal estimado con mínimos cuadrados ordinarios (MCO) es un modelo de caballo de batalla en Ciencias Políticas. Incluso cuando un académico utiliza un método más avanzado que puede hacer suposiciones más precisas sobre sus datos, como la regresión probit, un modelo de conteo o incluso un modelo bayesiano diseñado de manera única, el investigador a menudo se basa en la forma básica de un modelo que es lineal en los parámetros. De manera similar, muchos de los comandos R para estas técnicas más avanzadas utilizan una sintaxis funcional que se asemeja al código para estimar una regresión lineal. Por lo tanto, la comprensión de cómo usar R para estimar, interpretar y diagnosticar las propiedades de un modelo lineal se presta a un uso sofisticado de modelos con una estructura similar.

Este capítulo continúa describiendo el comando lm ( l inear m odel ) en R , que estima un modelo de regresión lineal con OLS, y las diversas opciones del comando. Luego, el capítulo describe cómo realizar diagnósticos de regresión de un modelo lineal. Estos diagnósticos sirven para evaluar si los supuestos críticos de la estimación de MCO se mantienen, o si nuestros resultados pueden estar sujetos a sesgos o ineficiencia.

A lo largo del capítulo, el ejemplo práctico es un análisis del número de horas que los profesores de biología de la escuela secundaria dedican a enseñar la evolución. El modelo replica el trabajo de Berkman y Plutzer (2010, Tabla 7.2 ), quienes argumentan que este resultado de política se ve afectado por factores a nivel estatal (como los estándares del plan de estudios) y los atributos de los maestros (como la capacitación). Los datos provienen de la Encuesta Nacional de Maestros de Biología de Escuelas Secundarias y consisten en 854 observaciones de maestros de biología de escuelas secundarias que fueron encuestados en la primavera de 2007. El resultado de interés es la cantidad de horas que un maestro dedica a la evolución humana y general en su o su clase de biología de la escuela secundaria ( hrs\_allev ), y las doce variables de entrada son las siguientes:  
fase 1: Un índice del rigor de los estándares de evolución de noveno y décimo grado en 2007 para el estado en el que trabaja el maestro. Esta variable está codificada en una escala estandarizada con media 0 y desviación estándar 1.

senior\_c: Una variable ordinal para la antigüedad del profesor. Codificado: 3 para 1 a 2 años de experiencia, 2 para 3 a 5 años, 1 para 6 a 10 años, 0 para 11 a 20 años y 1 para más de 21 años.

ph\_senior: Una interacción entre estándares y antigüedad.

notest\_p: Una variable indicadora codificada con 1 si el maestro informa que el estado no tiene una prueba de evaluación para la biología de la escuela secundaria, 0 si el estado tiene dicha prueba.

ph\_notest\_p: Una interacción entre estándares y ninguna prueba estatal.

mujer: Una variable indicadora codificada con 1 si el profesor es mujer, 0 si es hombre. Los valores que faltan se codifican 9.

biocred3: Una variable ordinal para la cantidad de horas crédito de biología que tiene el maestro (tanto de posgrado como de pregrado). Codificado 0 durante 24 horas o menos, 1 durante 25 a 40 horas y 2 durante más de 40 horas.

degr3: El número de títulos en ciencias que tiene el profesor, de 0 a 2.

evol\_course: Una variable indicadora codificada con 1 si el instructor tomó un curso específico de nivel universitario sobre evolución, 0 en caso contrario.

certificado: Un indicador codificado con 1 si el maestro tiene una certificación estatal normal, 0 en caso contrario.

idsci\_trans: Una medida compuesta, que va de 0 a 1, del grado en que el profesor se considera un científico.

seguro: Experiencia autoevaluada en teoría evolutiva. Codificado: 1 para “menos” que muchos otros maestros, 0 para “típico” de la mayoría de los maestros, 1 para “muy bueno” en comparación con la mayoría de los maestros de biología de la escuela secundaria y 2 para “excepcional” y a la par con los instructores de nivel universitario.

6.1 Estimación con mínimos cuadrados ordinarios Para empezar, necesitamos cargar los datos de la encuesta, a los que llamaremos evolución . En este ejemplo, cargamos un conjunto de datos con formato Stata. Esto es fácilmente posible a través de la biblioteca externa , que nos proporciona el comando read.dta : 1

rm (lista = ls ())

biblioteca (extranjera)

evolución <-read.dta (“BPchap7.dta,” convert.factors = FALSE)

Como regla general, queremos comenzar viendo las estadísticas descriptivas de nuestro conjunto de datos. Como mínimo, use el comando de resumen y quizás algunos de los otros comandos descritos en los Capítulos. 3 y 4:

resumen (evolución)

Además del resumen de estadísticas descriptivas que nos brinda, también enumerará el número de observaciones faltantes que tenemos sobre una variable dada (en NA ), si falta alguna. La condición predeterminada para la mayoría de los comandos de modelado en R es eliminar cualquier caso en el que falte una observación en cualquier variable de un modelo. Por lo tanto, el investigador debe ser consciente no solo de la variación en las variables relevantes, sino también de cuántos casos carecen de observación. 2 Además, los investigadores deben tener cuidado de notar cualquier cosa en las estadísticas descriptivas que se desvíe de los valores de una variable que se enumeran en el libro de códigos. Por ejemplo, en este caso la variable mujer tiene un valor máximo de 9. Si sabemos por nuestro libro de códigos que 0 y 1 son los únicos valores observados válidos de esta variable, entonces sabemos que cualquier otra cosa es un código erróneo o (en este caso) un valor faltante.

Antes de continuar, necesitamos reclasificar las observaciones faltantes de la hembra :

evolución $ mujer [evolución $ mujer == 9] <- NA

resumen (evolución)

evolución <-subconjunto (evolución,! is.na (mujer))

Este comando recodifica solo los valores de hembra codificados como 9 como faltantes. Como muestra la siguiente llamada al resumen , los 13 valores codificados como 9 ahora se enumeran como faltantes, por lo que se omitirán automáticamente en nuestro análisis posterior. Para asegurarnos de que cualquier cálculo que hagamos se centre solo en las observaciones sobre las que ajustamos el modelo, subconjuntamos nuestros datos para excluir las observaciones que faltan. Como alternativa al uso de subconjunto aquí, si tuviéramos valores perdidos en múltiples variables, en su lugar podríamos haber querido escribir: evolution <-na.omit (evolution) .

Habiendo limpiado nuestros datos, pasamos ahora al modelo de horas dedicadas a la enseñanza de la evolución descrito al comienzo del capítulo. Estimamos nuestro modelo lineal usando OLS:

mod.horas <-lm (hrs\_allev ~ fase1 \* senior\_c + fase1 \* notest\_p +

female + biocred3 + degr3 + evol\_course + certificada + idsci\_trans +  
  
 seguro, datos = evolución)

resumen (horas mod.)

La sintaxis estándar para especificar la fórmula de un modelo es enumerar la variable de resultado a la izquierda de la tilde ( ~ ) y las variables de entrada en el lado derecho separadas por signos más. Tenga en cuenta que incluimos dos términos especiales: fase1 \* senior\_c y fase1 \* notest\_p . Considerando el primero, fase1 \* senior\_c , esta notación interactiva agrega tres términos a nuestro modelo: fase1 , senior\_c y el producto de los dos. Estos modelos interactivos permiten efectos condicionales de una variable. 3 La opción de datos de lmnos permite llamar a variables del mismo conjunto de datos sin tener que hacer referencia al nombre del conjunto de datos con cada variable. Otras opciones destacadas para el comando lm incluyen subconjunto , que permite al usuario analizar solo una parte de un conjunto de datos, y ponderaciones , lo que permite al usuario estimar un modelo lineal con mínimos cuadrados ponderados (WLS). Observe que tuvimos que nombrar nuestro modelo en la estimación, llamándolo mod.hours por elección, y para obtener los resultados de nuestra estimación, necesitamos llamar a nuestro modelo con el comando de resumen . La salida del resumen (horas mod.) Se ve así:

Llamada:

lm (fórmula = hrs\_allev ~ fase1 \* senior\_c + fase1 \* notest\_p +

female + biocred3 + degr3 + evol\_course + certificada + idsci  
  
 \_trans +  
  
seguro, datos = evolución)

Derechos residuales de autor:

Mín. 1T Mediana 3T Máx.

-20,378 -6,148 -1,314 4,744 32,148

Coeficientes:

Estimar Std. Valor t de error Pr (> | t |)

(Intercepción) 10.2313 1.1905 8.594 <2e-16 \*\*\*

fase1 0,6285 0,3331 1,886 0,0596.

senior\_c -0.5813 0.3130 -1.857 0.0636.

notest\_p 0,4852 0,7222 0,672 0,5019

mujer -1,3546 0,6016 -2,252 0,0246 \*

biocred3 0.5559 0.5072 1.096 0.2734

degr3 -0.4003 0.3922 -1.021 0.3077

evol\_course 2.5108 0.6300 3.985 7.33e-05 \*\*\*

certificado -0,4446 0,7212 -0,617 0,5377

idsci\_trans 1.8549 1.1255 1.648 0.0997.

seguro 2.6262 0.4501 5.835 7.71e-09 \*\*\*

fase1: senior\_c -0.5112 0.2717 -1.881 0.0603.

fase1: notest\_p -0.5362 0.6233 -0.860 0.3899

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05. 0,1 1

Error estándar residual: 8,397 en 828 grados de libertad

R cuadrado múltiple: 0,1226, R cuadrado ajustado: 0,1099

Estadístico F: 9.641 en 12 y 828 DF, valor de p: <2.2e-16

La parte superior de la impresión repite el comando del modelo especificado por el usuario y luego proporciona algunas estadísticas descriptivas para los residuos. La siguiente tabla presenta los resultados de interés principal: La primera columna enumera todos los predictores del modelo, incluida una intersección. La segunda columna presenta la estimación MCO del coeficiente de regresión parcial. La tercera columna presenta la relación t para una hipótesis nula de que el coeficiente de regresión parcial es cero, y la cuarta columna presenta un valor p de dos colas para la relación t . Finalmente, la tabla imprime puntos y estrellas en función de los umbrales que cruza el valor p de dos colas . 4Debajo de la tabla, se informan varios estadísticos de ajuste: el error estándar de regresión (o error estándar residual), los valores R 2 y R 2 ajustados , y la prueba F para determinar si el modelo en su conjunto explica una parte significativa de la varianza. Los resultados de este modelo también se presentan de manera más formal en la Tabla 6.1 . 5 Cuadro 6.1 Modelo lineal de horas de clase dedicadas a la evolución de la enseñanza por profesores de biología de secundaria (estimaciones de OLS)

Vaticinador

Estimar

Std. Error

valor t

Pr (> |  t  |)

Interceptar

10.2313

1.1905

8.59

0,0000

Índice de estándares 2007

0,6285

0.3331

1,89

0.0596

Antigüedad (centrada)

-0,5813

0.3130

-1,86

0.0636

Estándares × antigüedad

-0,5112

0.2717

-1,88

0.0603

Cree que no hay prueba

0.4852

0,7222

0,67

0.5019

Estándares × cree que no hay prueba

-0,5362

0,6233

-0,86

0.3899

La maestra es mujer

-1,3546

0,6016

-2,25

0.0246

Créditos obtenidos en biología (0-2)

0.5559

0.5072

1,10

0.2734

Grados en ciencias (0-2)

-0.4003

0.3922

-1.02

0.3077

Clase de evolución completada

2.5108

0,6300

3,99

0,0001

Tiene certificación normal

-0,4446

0,7212

-0,62

0.5377

Se identifica como científico

1.8549

1.1255

1,65

0.0997

Experiencia autoevaluada (- 1 a +2)

2.6262

0.4501

5,84

0,0000

Notas : N  = 841. R 2  = 0. 1226. F 12, 828  = 9. 641 ( p  <0. 001). Datos de Berkman y Plutzer (2010)

Muchos investigadores, en lugar de informar las relaciones t y los valores p presentados en la salida predeterminada de lm , informarán los intervalos de confianza de sus estimaciones. Se debe tener cuidado en la interpretación de los intervalos de confianza, por lo que se insta a los lectores que no estén familiarizados con estos a consultar un libro de texto de estadística o econometría para obtener más información (como Gujarati y Porter 2009, págs. 108-109). Para construir tal conf idencia int Erval en R , el usuario debe elegir un nivel de confianza y el uso de la CONFINT comando:

confint (mod. horas, nivel = 0,90)

La opción de nivel es donde el usuario especifica el nivel de confianza. 0,90 corresponde al 90% de confianza, mientras que el nivel = 0,99 , por ejemplo, produciría un intervalo de confianza del 99%. Los resultados de nuestro intervalo de confianza del 90% se informan de la siguiente manera:

5% 95%

(Intercepción) 8.27092375 12.19176909

fase1 0.07987796 1.17702352

senior\_c -1.09665413 -0.06587642

notest\_p -0.70400967 1.67437410

mujer -2,34534464 -0,36388231

biocred3 -0,27927088 1,39099719

degr3 -1.04614354 0.24552777

evol\_course 1.47336072 3.54819493

certificado -1.63229086 0.74299337

idsci\_trans 0.00154974 3.70834835

seguro 1.88506881 3.36729476

fase1: senior\_c -0.95856134 -0.06377716

fase1: notest\_p -1.56260919 0.49020149

Entre otras características, un atributo útil de estos es que un lector puede examinar un intervalo de confianza del 90% (por ejemplo) y rechazar cualquier hipótesis nula que proponga un valor fuera del rango del intervalo para una prueba de dos colas. Por ejemplo, el intervalo para la variable seguro no incluye cero, por lo que podemos concluir con un 90% de confianza que el coeficiente parcial de esta variable es diferente de cero. 6

6.2 Diagnóstico de regresión Solo nos contentamos con usar MCO para estimar un modelo lineal si es el mejor estimador lineal imparcial (AZUL). En otras palabras, queremos obtener estimaciones que, en promedio, produzcan el verdadero parámetro de población (insesgado), y entre los estimadores insesgados queremos el estimador que minimiza la varianza del error de nuestras estimaciones (mejor o eficiente). Según el teorema de Gauss-Markov, MCO es AZUL y válido para inferencias si se cumplen cuatro supuestos:

1. Valores de entrada fijos o exógenos. En otras palabras, los predictores ( X ) deben ser independientes del término de error. Cov (X2 yo,tuI) = Cov (X3 yo,tuI) = ⋯ = Cov (Xk yo,tuI) = 0 .
2. Forma funcional correcta. En otras palabras, la media condicional de la perturbación debe ser cero.

E ( u yo  |  X 2 yo ,  X 3 yo ,  … ,  X ki ) = 0.

1. Homoscedasticidad o varianza constante de las perturbaciones ( u i ). Var ( u i ) =  σ 2 .
2. No existe autocorrelación entre perturbaciones. Cov ( u i ,  u j ) = 0 para i ≠ j .

Si bien nunca observamos los valores de las perturbaciones, ya que estos son términos de población, podemos predecir los residuos ( tu^ ) después de estimar un modelo lineal. Por lo tanto, normalmente usaremos residuos para evaluar si estamos dispuestos a hacer los supuestos de Gauss-Markov. En las siguientes subsecciones, realizamos diagnósticos de regresión para evaluar los diversos supuestos y describir cómo podríamos llevar a cabo medidas correctivas en R para corregir las aparentes violaciones de los supuestos de Gauss-Markov. La única excepción es que no probamos el supuesto de no autocorrelación porque no podemos hacer referencia a nuestros datos de ejemplo por tiempo o espacio. Ver cap. 9 para ver ejemplos de pruebas y correcciones de autocorrelación. Además, describimos cómo diagnosticar si los errores tienen una distribución normal, que es esencial para la inferencia estadística. Por último, consideramos la presencia de dos características de datos notables (multicolinealidad y observaciones atípicas) que no forman parte de los supuestos de Gauss-Markov pero que, no obstante, vale la pena comprobar.

6.2.1 Forma funcional Es fundamental tener la forma funcional correcta en un modelo lineal; de lo contrario, sus resultados estarán sesgados . Por lo tanto, al estimar un modelo lineal, debemos evaluar si hemos especificado el modelo correctamente o si debemos incluir aspectos no lineales de nuestros predictores (como logaritmos, raíces cuadradas, cuadrados, cubos o splines). Como regla general, un diagnóstico esencial para cualquier modelo lineal es hacer un diagrama de dispersión de los residuos ( tu^ ). Estos gráficos deben realizarse contra ambos valores ajustados ( Y^ ) y contra los predictores ( X ). Para construir una gráfica de residuos contra valores ajustados, simplemente haríamos referencia a los atributos del modelo que estimamos en una llamada al comando plot :

plot (y = mod.horas $ residuales, x = mod.horas $ valores ajustados,

xlab = "Valores ajustados", ylab = "Residuales")

Observe que los residuales de $ mod.horas nos permitieron hacer referencia a los residuos del modelo ( tu^ ) y mod.hours $ fit.values nos permitieron llamar a los valores predichos ( Y^ ). Podemos hacer referencia a muchas funciones con el signo de dólar ( $ ). Escriba nombres (mod.horas) para ver todo lo que se guarda. Volviendo a nuestro gráfico de salida, se presenta en la figura  6.1 . Como analistas, deberíamos comprobar este gráfico en busca de algunas características: ¿El promedio local de los residuos tiende a permanecer alrededor de cero? Si los residuos muestran un patrón claro de aumento o disminución en cualquier rango, entonces la forma funcional de alguna variable puede ser incorrecta. ¿El margen de los residuos difiere en alguna parte del gráfico? Si es así, puede haber un problema de heterocedasticidad. Una característica aparente de la Fig.  6.1es que los residuos parecen golpear un “piso” diagonal cerca del fondo de la nube. Esto surge porque un profesor no puede dedicar menos de cero horas a la enseñanza de la evolución. Por lo tanto, este piso natural refleja un límite en la variable dependiente. Una limitación de forma funcional como esta a menudo se aborda mejor dentro del marco del modelo lineal generalizado, que se considerará en el próximo capítulo. Abrir imagen en nueva ventanaFigura 6.1 Figura 6.1 Diagrama de dispersión de residuos frente a valores ajustados del modelo de evolución de las horas de enseñanza

Otra herramienta útil es dibujar cifras de los residuos frente a uno o más predictores. La figura 6.2 muestra dos gráficas de los residuos de nuestro modelo frente a la escala compuesta del grado en que el profesor se identifica a sí mismo como científico. La figura 6.2a muestra la gráfica básica usando los datos brutos, que un investigador siempre debe mirar. En este caso, el predictor de interés toma 82 valores únicos, pero muchas observaciones toman los mismos valores, particularmente en el extremo superior de la escala. En casos como este, muchos puntos de la trama se superpondrán entre sí. Al alterar los valores de idsci\_trans , o agregar un pequeño número extraído al azar, es más fácil ver dónde está la preponderancia de los datos. Figura 6.2 b muestra una gráfica revisada que hace temblar al predictor. El riesgo de la figura nerviosa es que mover los datos puede distorsionar un patrón real entre el predictor y los residuales. Sin embargo, en el caso de una variable de entrada ordinal (o quizás semi-ordinal), las dos subfiguras pueden complementarse entre sí para ofrecer la imagen más completa posible. Los dos diagramas de dispersión de la figura  6.2 se generan de la siguiente manera: Abrir imagen en nueva ventanaFigura 6.2 Figura 6.2 Diagrama de dispersión de residuos contra el grado en que un profesor se identifica como científico. ( a ) Datos brutos. ( b ) Datos alterados

plot (y = mod.horas $ residuales, x = evolución $ idsci\_trans,

xlab = "Se identifica como científico", ylab = "Residuales")

plot (y = mod.horas $ residuales, x = jitter (evolución $ idsci\_trans,

cantidad = .01), xlab = "Se identifica como científico (nervioso)",  
  
 ylab = "Residuales")

Al igual que la gráfica de residuo a valor ajustado de la figura  6.1 , examinamos las gráficas de residuo a predictor de la figura  6.2 para ver los cambios en la media local, así como las diferencias en la dispersión de los residuos, cada una de las cuales depende del predictor valor. En la forma funcional, hay pocos indicios de que la media móvil esté cambiando marcadamente entre los valores. Por lo tanto, al igual que con el gráfico de residuo a ajustado, vemos poca necesidad de volver a especificar nuestro modelo con una versión no lineal de este predictor. Sin embargo, la propagación de los residuos parece un poco preocupante, por lo que volveremos a tratar este tema en la siguiente sección.

Además de los métodos gráficos, una estadística de prueba común para diagnosticar una forma funcional mal especificada es la prueba RESET de Ramsey (prueba de error de especificación de regresión). Esta prueba procede reestimando el modelo original, pero esta vez incluyendo los valores ajustados del modelo original en alguna forma no lineal (como una fórmula cuadrática o cúbica). El uso de una relación F para evaluar si el nuevo modelo explica una varianza significativamente mayor que el modelo anterior sirve como prueba de si se debe incluir una forma diferente de uno o más predictores en el modelo. Podemos realizar esta prueba para una forma funcional cúbica potencial de la siguiente manera:

evolución $ ajuste <-mod.horas $ valores ajustados

reset.mod <-lm (hrs\_allev ~ fase1 \* senior\_c + fase1 \* notest\_p +

female + biocred3 + degr3 + evol\_course + certificada + idsci\_trans +  
  
 seguro + I (ajuste ^ 2) + I (ajuste ^ 3), datos = evolución)

anova (mod.horas, reset.mod)

La primera línea de código guarda los valores ajustados del modelo original como una variable en el marco de datos. La segunda línea agrega formas al cuadrado y al cubo de los valores ajustados en el modelo de regresión. Al incorporar estos términos dentro de la función I (nuevamente significando, “como i s”), podemos transformar algebraicamente la variable de entrada sobre la marcha mientras estimamos el modelo. En tercer lugar, el ANOVA de comandos (por un ÁLISIS o f va riance) presenta los resultados de un F -test que compara el modelo original para el modelo incluyendo una forma cuadrática y cúbica de los valores ajustados. En este caso, obtenemos un resultado de F 2826  = 2. 5626, con una p-valor de p  = 0. 07772. Esto indica que el modelo con el polinomio cúbico de valores ajustados se ajusta significativamente mejor al nivel del 90%, lo que implica que otra forma funcional sería mejor.

Para determinar qué predictor podría ser el culpable de la forma funcional mal especificada, podemos realizar pruebas de Durbin-Watson en los residuos, clasificando el predictor que puede ser problemático. ( Tenga en cuenta que, tradicionalmente, las pruebas de Durbin-Watson se clasifican a tiempo para probar la autocorrelación temporal. Esta idea se revisa en el capítulo  9 ). Un resultado discernible indica que los residuos toman valores similares en valores similares de la entrada, un signo de que el predictor necesita ser respecificado. La biblioteca lmtest (los usuarios necesitarán instalar con install.packages la primera vez) proporciona comandos para varias pruebas de diagnóstico, incluida la prueba Durbin-Watson. Clasificación de los residuos según el rigor de los estándares de evolución ( fase 1 ), ejecutamos la prueba:

install.packages (“lmtest”)

biblioteca (lmtest)

dwtest (mod.horas, order.by = evolución $ fase1)

Esto produce un estadístico de d  = 1. 8519 con un valor p aproximado de p  = 0. 01368, lo que indica que los residuos son similares según el valor de la covariable. Por lo tanto, podríamos proceder a volver a cifrar nuestra forma funcional agregando términos polinomiales para la fase1 :

mod.cubic <-lm (hrs\_allev ~ fase1 \* senior\_c + fase1 \* notest\_p +

female + biocred3 + degr3 + evol\_course + certificada + idsci\_trans +  
  
 seguro + I (fase1 ^ 2) \* senior\_c + I (fase1 ^ 3) \* senior\_c +  
  
 I (fase1 ^ 2) \* notest\_p + I (fase1 ^ 3) \* notest\_p, datos = evolución)

Al igual que con la prueba RESET en sí, nuestro nuevo modelo ( mod.cubic ) ilustra cómo podemos usar características adicionales del comando lm . Nuevamente, al usar la función I , podemos realizar álgebra en cualquier variable de entrada dentro del comando del modelo. Como antes, el signo de intercalación ( ^ ) eleva una variable a una potencia, lo que permite nuestra función polinomial. Nuevamente, para los términos de interacción, simplemente multiplicar dos variables con un asterisco ( \* ) asegura que se incluyan los efectos principales y los términos del producto de todas las variables en la interacción. Por lo tanto, permitimos que la antigüedad y si no hay una prueba de evaluación interactúen con la forma polinomial completa de los estándares de evolución.

6.2.2 Heteroscedasticidad Cuando la varianza del error en los residuos no es uniforme en todas las observaciones, un modelo tiene una varianza del error heterocedástica, las estimaciones son ineficientes y los errores estándar están sesgados para ser demasiado pequeños. La primera herramienta que usamos para evaluar si la varianza del error es homocedástica (o constante para todas las observaciones) versus heterocedástica es una gráfica de dispersión simple de los residuos. La figura 6.1 nos ofreció la gráfica de nuestros residuales contra los valores ajustados, y la figura  6.2ofrece un gráfico de ejemplo de los residuos frente a un predictor. Además de estudiar la media móvil para evaluar la forma funcional, también evaluamos la dispersión de los residuos. Si la dispersión de los residuos es una banda constante alrededor de cero, entonces podemos usar esto como una confirmación visual de la homocedasticidad. Sin embargo, en los dos paneles de la figura  6.2 , podemos ver que la preponderancia de los residuos se concentra más estrechamente cerca de cero para los profesores que están menos inclinados a identificarse a sí mismos como científicos, mientras que los residuos están más dispersos entre aquellos. que están más inclinados a identificarse como científicos. (Los residuos extremos son aproximadamente los mismos para todos los valores de X, lo que hace que sea un poco más difícil de detectar, pero la dispersión de puntos de datos concentrados en el medio se expande a valores más altos). Todo esto sugiere que la autoidentificación como científico se corresponde con la heterocedasticidad para este modelo.

Además de los métodos visuales, también tenemos la opción de utilizar una estadística de prueba en una prueba de Breusch-Pagan. Usando la biblioteca lmtest (que cargamos anteriormente), la sintaxis es la siguiente:

bptest (mod.horas, studentize = FALSE)

El valor predeterminado de bptest es utilizar la versión estudiantil de Koenker de esta prueba. Por lo tanto, la opción studentize = FALSE le da al usuario la opción de usar la versión original de la prueba Breusch-Pagan. La hipótesis nula en esta prueba de chi-cuadrado es la homocedasticidad. En este caso, nuestra estadística de prueba es χ212 d f= 51,7389 ( p  <0. 0001). Por lo tanto, rechazamos la hipótesis nula y concluimos que los residuos no son homocedásticos.

Sin homocedasticidad, nuestros resultados no son eficientes, entonces, ¿cómo podríamos corregir esto? Quizás la solución más común a este problema es utilizar errores estándar robustos de Huber-White o errores estándar tipo sándwich (Huber 1967; blanco 1980). La desventaja de este método es que ignora la ineficiencia de las estimaciones de MCO y continúa reportándolas como estimaciones de los parámetros. La ventaja, sin embargo, es que aunque las estimaciones de MCO son ineficientes bajo heterocedasticidad, son insesgadas. Dado que los errores estándar están sesgados, corregirlos soluciona el mayor problema que nos presenta la heterocedasticidad. Se pueden calcular los errores estándar de Huber-White utilizando las bibliotecas sándwich (que necesitan una primera instalación) y lmtest :

install.packages (“sándwich”)

biblioteca (sándwich)

coeftest (horas mod., vcov = vcovHC)

La biblioteca lmtest hace que el comando coeftest esté disponible, y la biblioteca sándwich hace que la matriz de varianza-covarianza vcovHC esté disponible dentro de esta. (Ambas bibliotecas requieren instalación en el primer uso). El comando coeftest presentará ahora los resultados de mod.hours nuevamente, con las mismas estimaciones de OLS que antes, los nuevos errores estándar de Huber-White y los valores de t y p que corresponden a la nuevos errores estándar.

Finalmente, también tenemos la opción de volver a estimar nuestro modelo usando WLS. Para hacer esto, el analista debe construir un modelo de los residuos cuadrados como una forma de pronosticar la varianza del error heterocedástico para cada observación. Si bien hay algunas formas de hacer esto de manera efectiva, aquí está el código para un plan. Primero, guardamos los residuos al cuadrado y ajustamos un modelo auxiliar del logaritmo de estos residuos al cuadrado:

evolución $ resid2 <-mod.horas $ residuales ^ 2

peso.reg <-lm (log (resid2) ~ fase1 \* senior\_c + fase1 \* notest\_p +

female + biocred3 + degr3 + evol\_course + certificada + idsci\_trans +  
  
 seguro, datos = evolución)

Una advertencia clave de WLS es que todos los pesos deben ser no negativos. Para garantizar esto, el código aquí modela el logaritmo de los residuos al cuadrado; por lo tanto, el exponencial de los valores ajustados de esta regresión auxiliar sirve como predicciones positivas de los residuos al cuadrado. (También existen otras soluciones a este problema). La regresión auxiliar simplemente incluye todos los predictores de la regresión original en su forma lineal, pero el usuario no está vinculado a esta suposición. De hecho, WLS ofrece el AZUL bajo heterocedasticidad, pero solo si el investigador modela correctamente la varianza del error. Por tanto, la especificación adecuada de la regresión auxiliar es esencial. En WLS, esencialmente queremos ponderar valores con una varianza de error baja y dar poca importancia a aquellos con una varianza de error alta. Por lo tanto, para nuestra regresión WLS final, elEl comando de pesos toma el recíproco de los valores predichos (exponenciados para estar en la escala original de los residuos al cuadrado):

wls.mod <-lm (hrs\_allev ~ fase1 \* senior\_c + fase1 \* notest\_p +

female + biocred3 + degr3 + evol\_course + certificada + idsci\_trans +  
  
 seguro, datos = evolución,  
  
 pesos = I (1 / exp (peso.reg $ valores ajustados)))

resumen (wls.mod)

Esto nos presenta un conjunto de estimaciones que explica la heterocedasticidad en los residuos.

6.2.3 Normalidad Si bien no forma parte del teorema de Gauss-Markov, una suposición importante que hacemos con los modelos de regresión lineal es que las perturbaciones se distribuyen normalmente. Si esta suposición no es cierta, entonces OLS sigue siendo AZUL. Sin embargo, el supuesto de normalidad es esencial para que nuestras estadísticas inferenciales habituales sean precisas. Por lo tanto, probamos este supuesto examinando la distribución empírica de los residuos predichos. Un primer lugar fácil para comenzar es examinar un histograma de los residuos.

hist (mod.horas $ residuales, xlab = “Residuales,” main = "")

Este histograma se muestra en la figura  6.3 . Generalmente, nos gustaría una curva de campana simétrica que no sea ni excesivamente plana ni puntiaguda. Si tanto el sesgo (refiriéndose a si la distribución es simétrica o si las colas son pares) como la curtosis (refiriéndose al pico de la distribución) son similares a una distribución normal, podemos usar esta figura a favor de nuestra suposición. En este caso, los residuos parecen estar sesgados a la derecha, lo que sugiere que la normalidad no es un supuesto seguro en este caso. Abrir imagen en nueva ventanaFigura 6.3 Figura 6.3 Histograma de residuos del modelo de evolución de horas lectivas

Una figura un poco más compleja (aunque potencialmente más informativa) se denomina gráfico de cuantiles-cuantiles. En esta figura, los cuantiles de los valores empíricos de los residuos se grafican contra los cuantiles de una distribución normal teórica. Cuanto menos correspondan estas cantidades, menos razonable es suponer que los residuos se distribuyen normalmente. Tal figura se construye en R de la siguiente manera:

qqnorm (horas mod. $ residuales)

qqline (mod.horas $ residuales, col = “rojo”)

La primera línea de código ( qqnorm ) en realidad crea el gráfico de cuantiles-cuantiles. La segunda línea ( qqline ) agrega una línea guía a la parcela existente. El gráfico completo se encuentra en la Fig.  6.4 . Como puede verse, en cuantiles inferiores y superiores, los valores de la muestra se desvían sustancialmente de los valores teóricos. Una vez más, esta figura cuestiona el supuesto de normalidad. Abrir imagen en nueva ventanaFigura 6.4 Figura 6.4 Gráfico de cuantiles-cuantiles normales para los residuos del modelo de evolución de las horas de enseñanza

Además de estas evaluaciones sustancialmente enfocadas de la distribución empírica, los investigadores también pueden usar estadísticas de prueba. El estadístico de prueba más comúnmente utilizado en este caso es la prueba de Jarque-Bera, que se basa en el sesgo y la curtosis de la distribución empírica de los residuos. Esta prueba utiliza la hipótesis nula de que los residuos están distribuidos normalmente y la hipótesis alternativa de que no. 7 La biblioteca tseries puede calcular esta estadística, que instalamos en el primer uso:

install.packages (“tseries”)

biblioteca (tseries)

jarque.bera.test (mod.horas $ residuales)

En nuestro caso, χ 2  = 191. 5709, por lo que rechazamos la hipótesis nula y concluimos que los residuos no están distribuidos normalmente. Al igual que los diagnósticos de heterocedasticidad, preferiríamos un resultado nulo ya que preferimos no rechazar la suposición.

Los tres diagnósticos indican una violación del supuesto de normalidad, entonces, ¿cómo podríamos responder a esta violación? En muchos casos, la mejor respuesta probablemente se encuentre en el próximo capítulo sobre modelos lineales generalizados (GLM). Bajo este marco, podemos asumir una gama más amplia de distribuciones para la variable de resultado, y también podemos transformar la variable de resultado a través de una función de enlace. Otra opción algo similar sería transformar la variable dependiente de alguna manera. En el caso de nuestro ejemplo de ejecución sobre las horas dedicadas a la evolución, nuestra variable de resultado no puede ser negativa, por lo que podríamos agregar 1 a la respuesta de cada maestro y tomar el logaritmo de nuestra variable dependiente. Sin embargo, tenga en cuenta que esto tiene un mayor impacto en la forma funcional del modelo (ver Gujarati y Porter 2009, págs. 162-164), y tenemos que suponer que las perturbaciones del modelo con una variable dependiente registrada se distribuyen normalmente con fines inferenciales.

6.2.4 Multicolinealidad Aunque no es un supuesto estadístico del modelo lineal, ahora pasamos a diagnosticar la presencia de multicolinealidad entre predictores. Multicolinealidad significa que un predictor es una función de uno o más predictores. Si un predictor es una función exacta de otros predictores, entonces existe una multicolinealidad perfecta en el conjunto de regresores. En una multicolinealidad perfecta, el modelo no puede estimarse tal cual y debe volver a especificarse. Por ejemplo, si un investigador incluye tanto el “año de nacimiento” como la “edad” de un encuestado en un análisis transversal, una variable sería una función perfecta de la otra y, por lo tanto, el modelo no sería estimable.

Una situación común es que un predictor tenga una multicolinealidad alta, pero no perfecta. El problema que surge es que los errores estándar de los coeficientes de regresión comenzarán a aumentar. Sin embargo, es importante destacar que OLS sigue siendo AZUL en el caso de multicolinealidad alta pero imperfecta. En otras palabras, los grandes errores estándar son precisos y aún reflejan el estimador más eficiente posible. Sin embargo, a menudo es una buena idea tener una idea de si la multicolinealidad está presente en un modelo de regresión.

El enfoque general para evaluar la multicolinealidad se basa en regresiones auxiliares de predictores. Entre las medidas de resumen de estos resultados se encuentra el factor de inflación de la varianza (VIF). Para cada predictor, el VIF nos da una idea del grado en que la varianza común entre los predictores aumenta el error estándar del coeficiente del predictor. Los VIF pueden tomar cualquier valor no negativo, y los valores más pequeños son más deseables. Una regla general común es que siempre que un VIF excede 10, se puede concluir que la multicolinealidad está dando forma a los resultados. 8

En R , los VIF se pueden calcular para todos los coeficientes utilizando la biblioteca de automóviles , instalada en el Cap. 2 :

biblioteca (coche)

vif (horas mod.)

Los VIF calculados de esta manera se presentan en la Tabla 6.2 . Como se puede ver en la tabla, todos los VIF son pequeños, lo que implica que la multicolinealidad no es un problema importante en este modelo. Sin embargo, en situaciones en las que surge la multicolinealidad, a veces el mejor consejo es no hacer nada. Para una discusión sobre cómo decidir si no hacer nada es el mejor enfoque o si otra solución funcionaría mejor, consulte Gujarati y Porter (2009, págs. 342-346). Cuadro 6.2 Factores de inflación de la varianza para los predictores de la evolución de las horas dedicadas a la enseñanza

Vaticinador

VIF

Índice de estándares 2007

1,53

Antigüedad (centrada)

1.12

Estándares × antigüedad

1,10

Cree que no hay prueba

1.12

Estándares × cree que no hay prueba

1,63

La maestra es mujer

1.08

Créditos obtenidos en biología (0-2)

1,15

Grados en ciencias (0-2)

1,11

Clase de evolución completada

1,17

Tiene certificación normal

1.03

Se identifica como científico

1.12

Experiencia autoevaluada (- 1 a +2)

1,20

6.2.5 Valores atípicos, apalancamiento y puntos de datos influyentes Como diagnóstico final, es una buena idea determinar si alguna observación está ejerciendo una influencia excesiva en los resultados de un modelo lineal. Si una o dos observaciones generan un resultado completo que de otra manera no surgiría, entonces un modelo que incluya estas observaciones puede ser engañoso. Consideramos tres tipos de puntos de datos problemáticos: valores atípicos (para los cuales el residual es excesivamente grande), puntos de apalancamiento (que toman un valor de un predictor que está desproporcionadamente distante de otros valores) y puntos de influencia (valores atípicos con mucho apalancamiento). . Los más problemáticos son los puntos de influencia porque tienen la mayor capacidad para distorsionar los coeficientes de regresión parcial.

Un diagnóstico simple para estas características de las observaciones nuevamente es simplemente examinar diagramas de dispersión de residuos, como los que se informan en las Figs. 6.1 y 6.2 . Si una observación se destaca en la escala del predictor, entonces tiene apalancamiento. Si se destaca en la escala residual, entonces es un valor atípico. Si destaca en ambas dimensiones, entonces es un punto de influencia. Ninguna de las cifras de este modelo muestra señales de advertencia al respecto. Otra opción para evaluar estos atributos para las observaciones es calcular las cantidades de residuos estudentizados para detectar valores atípicos, valores de sombrero para detectar puntos de apalancamiento y distancias de Cook para detectar puntos de datos influyentes. La biblioteca de automóviles ofrece nuevamente una forma sencilla de ver estas cantidades para todas las observaciones.

influenceIndexPlot (mod.horas,

vars = c ("Cocinero", "Studentizado", "sombrero"), id.n = 5)

Los valores de estas tres cantidades se muestran en la figura 6.5 . , que muestra las distancias de Cook, los residuos estudentizados y los valores de sombrero, respectivamente. En cualquiera de estos gráficos, un valor extremo en relación con los demás indica que una observación puede ser particularmente problemática. En esta figura, ninguna de las observaciones se destaca particularmente, y ninguno de los valores de la distancia de Cook está ni remotamente cerca de 1 (que es un umbral común de regla empírica para esta cantidad). Por lo tanto, ninguna de las observaciones parece ser particularmente problemática para este modelo. En un caso en el que algunas observaciones parecen ejercer influencia sobre los resultados, el investigador debe decidir si es razonable mantener las observaciones en el análisis o si alguna de ellas debe eliminarse. La eliminación de datos de un modelo lineal se puede lograr fácilmente con la opción de subconjunto de lm. Abrir imagen en nueva ventanaFigura 6.5 Figura 6.5 Distancias de Cook, residuos estudentizados y valores de sombrero del modelo de evolución de las horas de enseñanza

Ahora hemos considerado cómo ajustar modelos lineales en R y cómo realizar varios diagnósticos para determinar si OLS nos presenta el AZUL. Si bien este es un modelo común en Ciencias Políticas, los investigadores con frecuencia necesitan modelar variables dependientes limitadas en el estudio de la política. Para abordar las variables dependientes de esta naturaleza, pasamos al capítulo siguiente a GLM. Estos modelos se basan en el marco del modelo lineal, pero permiten variables de resultado de naturaleza limitada o categórica.

6.3 Problemas de práctica Este conjunto de problemas de práctica se basará en la de Owsiak (2013) trabajo sobre democratización, en el que muestra que los estados que liquidan todas sus fronteras internacionales tienden a democratizarse. Cargue la biblioteca externa y luego descargue un subconjunto de los datos de Owsiak, guardados en el archivo con formato Stata owsiakJOP2013.dta . Este archivo se puede descargar del Dataverse vinculado en la página vii o del contenido del capítulo vinculado en la página 79. Estos son datos de panel que incluyen observaciones para 200 países desde 1918 hasta 2007, con un total de 10,434 países-año que forman los datos. Los países en estos datos cambian con el tiempo (tal como cambiaron en su libro de historia), lo que lo convierte en lo que llamamos un panel desequilibrado. Por lo tanto, nuestro modelo posterior incluye valores rezagados de varias variables o valores del año anterior. Ver cap. 8 para obtener más información sobre los datos anidados y el Cap. 9 para obtener más información sobre los datos temporales. Para este ejercicio, nuestras herramientas OLS estándar funcionarán bien.  
1. Comience usando el comando na.omit , que se describe en la página 81, para eliminar las observaciones faltantes de estos datos. Luego calcule las estadísticas descriptivas para las variables en este conjunto de datos.

1. Para replicar el Modelo 2 de Owsiak (2013), estimar una regresión lineal con MCO utilizando la siguiente especificación (con los nombres de las variables entre paréntesis): La variable dependiente es la puntuación de Polity ( polity2 ), y los predictores son un indicador de tener todas las fronteras establecidas ( allsettle ), PIB retrasado ( laggdpam ) , cambio rezagado en el PIB ( laggdpchg ), apertura comercial rezagada ( lagtradeopen ), personal militar rezagado ( lagmilper ), población urbana rezagada ( lagupop ), movimiento no democrático anterior rezagado ( lagsumdown ) y puntaje político rezagado ( lagpolity ).
2. Grafique los residuales contra los valores ajustados.
3. ¿Existe heterocedasticidad en los residuos? Basado en diagramas de dispersión y una prueba de Breusch-Pagan, ¿qué concluye? un. Estime los errores estándar de Huber-White para este modelo con la biblioteca sándwich y el comando coeftest .
4. Para obtener un crédito de bonificación , puede reproducir el de Owsiak (2013) resultados exactamente al calcular los errores estándar agrupados , agrupando por país (nombre de la variable: ccode ). Puede hacer esto en tres pasos: Primero, instale la biblioteca multiwayvcov . En segundo lugar, defina una matriz de varianza-covarianza de error mediante el comando cluster.vcov . En tercer lugar, utilice esa matriz de varianza-covarianza de error como argumento en el comando coeftest de la biblioteca lmtest .
5. Determine si la multicolinealidad es una preocupación calculando los VIF para los predictores en este modelo.
6. ¿Los residuos de este modelo están distribuidos normalmente? Utilice cualquiera de los métodos discutidos para sacar una conclusión.
7. Para el crédito de bonificación , puede evaluar si existe autocorrelación en los residuales, como se discutirá más adelante en el Cap. 9 . Para ello, primero instale la biblioteca plm . Segundo, reajuste su modelo usando el comando plm . (Asegúrese de especificar model = “pooling” como una opción en el comando para estimar con MCO). En tercer lugar, use la prueba pbgtest para realizar una prueba de panel de Breusch-Godfrey para evaluar si existe una correlación serial en los residuos. ¿Qué conclusión sacas?

Notas al pie 1 . El archivo de datos de Berkman y Plutzer, llamado BPchap7.dta , está disponible en el Dataverse vinculado en la página vii o en el contenido del capítulo vinculado en la página 79. Recuerde que es posible que deba usar el comando setwd para señalar dónde ha guardado los datos.

2 . Una alternativa teóricamente atractiva a la eliminación por listas como medio de manejar los datos faltantes es la imputación múltiple . Ver Little y Rubin (1987), Frotar (1987) y King et al. (2001) para más detalles.

3 . Ver Brambor et al. (2006) para obtener más detalles sobre los términos de interacción. Además, tenga en cuenta que se podría lograr una especificación equivalente de este modelo reemplazando phase1 \* senior\_c y phase1 \* notest\_p con los términos phase1 + senior\_c + ph\_senior + notest\_p + ph\_notest\_p . Simplemente estamos introduciendo cada uno de los términos por separado de esta manera.

4 . Se recuerda a los usuarios que para las pruebas de una cola, en las que el usuario desea probar que el coeficiente parcial específicamente es mayor o menor que cero, el valor p será diferente. Si el signo del coeficiente coincide con la hipótesis alternativa, entonces el valor p correspondiente es la mitad de lo que se informa. (Naturalmente, si el signo del coeficiente es opuesto al signo de la hipótesis alternativa, los datos no se ajustan a la hipótesis del investigador). Además, los investigadores pueden querer probar una hipótesis en la que la hipótesis nula sea algo diferente de cero: En En este caso, el usuario puede construir la relación t correcta utilizando la estimación informada y el error estándar.

5 . Los investigadores que escriben sus documentos con LaTeX pueden transferir fácilmente los resultados de un modelo lineal de R a una tabla usando la biblioteca xtable . (HTML también es compatible con xtable ). En el primer uso, instale con: install.packages (“xtable”) . Una vez instalado, simplemente ingrese a la biblioteca (xtable); xtable (mod.hours) produciría código listo para LaTeX para una tabla similar a la Tabla 6.1 . Como otra opción para generar resultados, consulte el paquete rtf sobre cómo generar resultados en formato de texto enriquecido.

6 . De hecho, también podríamos concluir que el coeficiente es mayor que cero al nivel de confianza del 95%. Para obtener más información sobre cómo los intervalos de confianza también pueden ser útiles para las pruebas de una cola, consulte Gujarati y Porter (2009, pag. 115).

7 . En otras palabras, si no rechazamos la hipótesis nula para una prueba de Jarque-Bera, entonces concluimos que no hay evidencia significativa de no normalidad. Tenga en cuenta que esto es diferente a concluir que tenemos normalidad. Sin embargo, esta es la conclusión más sólida que podemos sacar con esta estadística de prueba.

8 . Un VIF de 10 significa que el 90% de la varianza en un predictor puede explicarse por los otros predictores, lo que en la mayoría de los contextos puede considerarse como un gran grado de varianza común. Sin embargo, a diferencia de otras pruebas de diagnóstico, esta regla general no debe considerarse una estadística de prueba. En última instancia, el investigador debe sacar una conclusión sustancial de los resultados.

Material suplementario 318886\_1\_En\_6\_MOESM1\_ESM.zip (2.1 mb) Dataverse (2,154 KB) Referencias Berkman M, Plutzer E (2010) Evolución, creacionismo y la batalla por controlar las aulas de Estados Unidos. Cambridge University Press, Nueva York CrossRefGoogle Académico Brambor T, Clark WR, Golder M (2006) Comprensión de los modelos de interacción: mejora de los análisis empíricos. Polit Anal 14 (1): 63–82 CrossRefGoogle Académico Gujarati DN, Porter DC (2009) Econometría básica, 5ª ed. McGraw-Hill / Irwin, Nueva York Google Académico Huber PJ (1967) El comportamiento de las estimaciones de máxima verosimilitud en condiciones no estándar. En: LeCam LM, Neyman J (eds) Actas del quinto simposio de Berkeley sobre estadística matemática y probabilidad, volumen 1: estadística University of California Press, Berkeley, CA Google Académico King G, Honaker J, Joseph A, Scheve K (2001) Analizando datos incompletos de ciencia política: un algoritmo alternativo para la imputación múltiple. Am Polit Sci Rev 95 (1): 49–69 Google Académico Little RJA, Rubin DB (1987) Análisis estadístico con datos faltantes, 2ª ed. Wiley, Nueva York zbMATHGoogle Académico Owsiak AP (2013) Democratización y acuerdos fronterizos internacionales. J Polit 75 (3): 717–729 CrossRefGoogle Académico Rubin DB (1987) Imputación múltiple por falta de respuesta en encuestas. Wiley, Nueva York CrossRefzbMATHGoogle Académico White H (1980) Un estimador de matriz de covarianza consistente con heterocedasticidad y una prueba directa de heterocedasticidad. Econometrica 48 (4): 817–838 CrossRefMathSciNetzbMATHGoogle Académico

# 7 Modelos lineales generalizados

Palabras clave: - Criterio de información de Akaike - Modelo probit - Regresión binomial negativa - Regresión probit - Modelo binomial negativo

Si bien el modelo de regresión lineal es común a las ciencias políticas, muchas de las medidas de resultado que los investigadores desean estudiar son variables binarias, ordinales, nominales o de conteo. Cuando estudiamos estas variables dependientes limitadas, recurrimos a técnicas como regresión logística, regresión probit, regresión logit (y probit) ordenada, regresión logit (y probit) multinomial, regresión de Poisson y regresión binomial negativa. Se puede ver una revisión de estos y varios otros métodos en volúmenes como King (1989) y largo (1997).

De hecho, todas estas técnicas pueden considerarse como casos especiales del modelo lineal generalizado o GLM (Gill 2001). El enfoque GLM en resumen es transformar la media de nuestro resultado de alguna manera para que podamos aplicar la lógica habitual del modelado de regresión lineal a la media transformada. De esta manera, podemos modelar una amplia clase de variables dependientes para las cuales la distribución de los términos de perturbación viola el supuesto de normalidad del teorema de Gauss-Markov. Además, en muchos casos, el resultado está limitado, por lo que la función de enlace que usamos para transformar la media del resultado puede reflejar una forma funcional más realista (Gill 2001, págs. 31-32).

El comando glm en R es lo suficientemente flexible como para permitir al usuario especificar muchos de los GLM más utilizados, como la logística y la regresión de Poisson. Un puñado de modelos que se usan con cierta regularidad, como el logit ordenado y la regresión binomial negativa, en realidad requieren comandos únicos que también cubriremos. Sin embargo, en general, el comando glm es un buen lugar para buscar primero cuando un investigador tiene una variable dependiente limitada. De hecho, el comando glm toma un argumento llamado familia que permite al usuario especificar qué tipo de modelo desea estimar. Al escribir ? Familia en la R consola, el usuario puede obtener una descripción general rápida de qué modelos puede estimar el comando glm .

Este capítulo continúa ilustrando ejemplos con resultados binarios, resultados ordinales y resultados de conteo. Cada una de estas secciones utiliza un conjunto de datos de ejemplo diferente para considerar las variables dependientes de cada tipo. Cada sección presentará sus datos de ejemplo a su vez.

7.1 Resultados binarios Primero consideramos las variables de resultado binarias, o variables que toman solo dos valores posibles. Por lo general, estos resultados se codifican con 0 o 1 para simplificar la interpretación. Como ejemplo en esta sección, usamos datos de encuestas del Estudio Comparativo de Sistemas Electorales (CSES). Singh2014a) estudia un subconjunto de estos datos que consta de 44.897 encuestados de 30 elecciones. Estas elecciones ocurrieron entre los años 1996-2006 en los países de Canadá, República Checa, Dinamarca, Finlandia, Alemania, Hungría, Islandia, Irlanda, Israel, Italia, Países Bajos, Nueva Zelanda, Noruega, Polonia, Portugal, Eslovenia, España, Suecia, Suiza y Reino Unido.

Singh utiliza estos datos para evaluar cómo la distancia ideológica determina la elección de voto de las personas y su disposición a votar en primer lugar. Sobre la base del modelo espacial de la política propuesto por Hotelling (1929), Negro (1948), Downs (1957) y otros, el artículo muestra que las diferencias lineales en ideología explican mejor el comportamiento de los votantes que las diferencias cuadradas. Las variables del conjunto de datos son las siguientes: votó: Indicador codificado 1 si el encuestado votó, 0 si no.

votadoinc: Indicador codificado 1 si el encuestado votó por el partido en el poder, 0 si votó por otro partido. (Faltan los no votantes).

cntryyear: Una variable de carácter que enumera el país y el año de la elección.

cntryyearnum: Un índice numérico que identifica el país y el año de la elección.

distanciainc: Distancia entre el encuestado y el partido en el poder en una escala de ideología de 0 a 10.

distanciaincsq: Distancia al cuadrado entre el votante y el partido en el poder.

ponderado a distancia: Distancia entre el encuestado y el partido político más similar en una escala de ideología de 0 a 10, ponderada por la competitividad de la elección.

distanciasq ponderadas: Distancia ponderada al cuadrado entre el votante y el partido ideológico más similar.

Los datos se guardan en formato Stata, por lo que necesitaremos cargar la biblioteca externa . Descargue el archivo SinghJTP.dta del Dataverse vinculado en la página vii o el contenido del capítulo vinculado en la página 97. Luego abra los datos de la siguiente manera:

biblioteca (extranjera)

votando <-read.dta (“SinghJTP.dta,” convert.factors = FALSE)

Un buen paso inmediato aquí sería utilizar comandos como resumen y gráficos para tener una idea de los atributos descriptivos de los datos. Esto queda para el lector.

7.1.1 Modelos Logit Como primer modelo a partir de estos datos, modelaremos la probabilidad de que un encuestado haya votado por el partido en el poder, en lugar de por otro partido. Usaremos solo un predictor en este caso, y esa es la distancia entre el votante y el partido en el poder. Elaboramos esto como un modelo de regresión logística . La sintaxis de este modelo es:

inc. lineal <-glm (votadainc ~ distanciainc,

familia = binomio (enlace = "logit"), datos = votación)

La sintaxis de GLM ( g eneralized l inear m Odel) es casi idéntica a lm : Todavía empezamos con una especificación funcional que pone la variable dependiente a la izquierda de la tilde ( ~ ) y los predictores a la derecha separan con signos más. Nuevamente, hacemos referencia a nuestro conjunto de datos con la opción de datos . Ahora, sin embargo, debemos usar la opción de familia para especificar qué GLM queremos estimar. Al especificar binomial (link = “logit”) , declaramos un resultado binario y que estamos estimando un modelo logit, en lugar de probit. Después de la estimación, escribiendo resumen (incluido lineal) obtenemos el resultado de nuestro modelo de regresión logística, que es el siguiente:

Llamada:

glm (fórmula = votadainc ~ distanciainc, familia = binomial

(enlace = “logit”),

datos = votación)

Residuos de desviación:

Mín. 1T Mediana 3T Máx.

-1,2608 -0,8632 -0,5570 1,0962 2,7519

Coeficientes:

Estimar Std. Error z valor Pr (> | z |)

(Intercepción) 0.19396 0.01880 10.32 <2e-16 \*\*\*

distanciainc -0.49469 0.00847 -58.41 <2e-16 \*\*\*

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05. 0,1

1

(El parámetro de dispersión para la familia binomial se toma como 1)

Desviación nula: 47335 sobre 38210 grados de libertad

Desviación residual: 42910 sobre 38209 grados de libertad

(6686 observaciones eliminadas por falta de información)

AIC: 42914

Número de iteraciones de puntuación de Fisher: 4

La impresión es similar a la impresión del modelo lineal que estimamos en el Cap. 6 1 En la Tabla 7.1 se puede encontrar una presentación más formal de nuestros resultados . 2 Sin embargo, al comparar estos resultados con el modelo lineal, es importante tener en cuenta algunas diferencias. Primero, las estimaciones de los coeficientes en sí mismas no son tan significativas como las del modelo lineal. Un modelo logit transforma nuestro resultado de interés, la probabilidad de votar por el partido en el poder, porque está acotado entre 0 y 1. La transformada logit vale la pena porque nos permite usar un marco de predicción lineal, pero requiere un paso adicional de esfuerzo por la interpretación. (Ver secc.  7.1.3 para obtener más información sobre esto.) Una segunda diferencia en la salida es que informa las razones z en lugar de las razones t : al igual que antes, estas se calculan en torno a la hipótesis nula de que el coeficiente es cero, y la fórmula para la razón utiliza la estimación y error estándar de la misma manera. Sin embargo, ahora debemos asumir que estas estadísticas siguen una distribución normal, en lugar de una distribución t . 3 En tercer lugar, se presentan diferentes estadísticas de ajuste: puntuaciones de desviación y el criterio de información de Akaike (AIC). 4 Cuadro 7.1 Modelo logit de probabilidad de votar por el partido en el poder, 30 elecciones internacionales

Vaticinador

Estimar

Std. error

valor z

Pr (> |  z  |)

Interceptar

0,1940

0.0188

10,32

0,0000

Distancia

−0,4947

0,0085

−58,41

0,0000

Notas : N  = 38, 211. AIC = 42, 914. El 69% predijo correctamente. Datos de Singh (2014a)

En la tabla 7.1 , informamos los coeficientes, los errores estándar y la información inferencial. También reportamos el AIC, que es un buen índice de ajuste y tiene la característica de penalizar por el número de parámetros. A diferencia de R 2 en la regresión lineal, sin embargo, la AIC no tiene métrica natural que da un sentido absoluto de ajuste del modelo. Más bien, funciona mejor como un medio para comparar modelos, con valores más bajos que indican un mejor ajuste penalizado. Para incluir una medida de ajuste que tenga una escala natural, también informamos qué porcentaje de respuestas predice correctamente nuestro modelo. Para calcular esto, todo lo que necesitamos hacer es determinar si el modelo predeciría un voto para el partido en el poder y compararlo con cómo votó realmente el encuestado. En R, podemos rodar nuestro propio cálculo:

predicho <-como numérico (

predecir.glm (incluido lineal, tipo = "respuesta")>. 5)

verdadero <-voting $ voteinc [vote $ votó == 1]

correcto <-as.numeric (predicho == true)

100 \* tabla (correcta) / suma (tabla (correcta))

En la primera línea, creamos un vector de las predicciones del modelo. El código utiliza el comando predict.glm , que puede pronosticar de manera útil a partir de cualquier modelo estimado con el comando glm . Al especificar type = “response” , aclaramos que queremos que nuestras predicciones estén en la escala de probabilidad (en lugar de la escala predeterminada de utilidad latente). Luego preguntamos si cada probabilidad es mayor que 0.5. Envolviendo todo esto en el as.numericcomando, contamos todas las probabilidades por encima de 0,5 como valores predichos de 1 (para el titular) y todos los que son inferiores a 0,5 como valores previstos de 0 (contra el titular). En la segunda línea, simplemente subconjuntamos el vector original del resultado de los datos originales a los que votaron y por lo tanto se incluyeron en el modelo. Este paso de subconjunto es esencial porque el comando glm borra automáticamente los datos faltantes de la estimación. Por lo tanto, sin subconjuntos, nuestros valores predichos y verdaderos no se vincularían adecuadamente. En la tercera línea, creamos un vector codificado 1 si el valor predicho coincide con el valor verdadero, y en la cuarta línea creamos una tabla de este vector. La impresión es:

correcto

0 1

30.99108 69.00892

Por lo tanto, sabemos que el modelo predice correctamente el 69% de los valores de resultado, que informamos en la Tabla 7.1 .

Como un ejemplo más de regresión logística, Singh (2014a) compara un modelo con distancias ideológicas lineales con uno con distancias ideológicas cuadradas. Para ajustar este modelo alternativo, escribimos:

inc. al cuadrado <-glm (votadainc ~ distanciaincsq,

familia = binomio (enlace = "logit"), datos = votación)

resumen (incluido cuadrado)

El resultado del comando de resumen en la segunda línea es:

Llamada:

glm (fórmula = votadainc ~ distanciaincsq, familia = binomi

al (enlace = “logit”),

datos = votación)

Residuos de desviación:

Mín. 1T Mediana 3T Máx.

-1.1020 -0.9407 -0.5519 1.2547 3.6552

Coeficientes:

Estimar Std. Error z valor Pr (> | z |)

(Intercepción) -0.179971 0.014803 -12.16 <2e-16 \*\*\*

distanciaincsq -0.101549 0.002075 -48.94 <2e-16 \*\*\*

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05.

0,1 1

(El parámetro de dispersión para la familia binomial se toma como 1)

Desviación nula: 47335 sobre 38210 grados de libertad

Desviación residual: 43087 sobre 38209 grados de libertad

(6686 observaciones eliminadas por falta de información)

AIC: 43091

Número de iteraciones de puntuación de Fisher:} 5

Con este segundo modelo podemos ver cómo el AIC puede ser útil: Con un valor mayor de 43.091 en el modelo cuadrático, concluimos que el modelo con distancia ideológica lineal encaja mejor con un AIC de 42.914. Esto corresponde a la conclusión del artículo original de que la forma lineal de la variable se ajusta mejor.

7.1.2 Modelos Probit Los modelos Logit han ganado fuerza a lo largo de los años en aras de la simplicidad en el cálculo y la interpretación. (Por ejemplo, los modelos logit se pueden interpretar con razones de probabilidades ). Sin embargo, un supuesto clave de los modelos logit es que el término de error en el modelo de variable latente (o la utilidad latente) tiene una distribución logística. Podemos estar más contentos con suponer que el término de error de nuestro modelo está distribuido normalmente, dada la prevalencia de esta distribución en la naturaleza y en los resultados asintóticos. 5 La regresión probit nos permite ajustar un modelo con una variable de resultado binaria con un término de error distribuido normalmente en el modelo de variable latente.

Para mostrar cómo funciona este modelo alternativo de resultado binario, recurrimos a un modelo de probabilidad de que un encuestado haya votado. Singh2014a) modela esto en función de la proximidad ideológica al partido más cercano ponderada por la competitividad de la elección. La teoría aquí es que los individuos con una alternativa relativamente próxima en una elección competitiva tienen más probabilidades de considerar que vale la pena votar. Encajamos este modelo de la siguiente manera:

turnout.linear <-glm (votado ~ ponderado a distancia,

familia = binomio (enlace = "probit"), datos = votación)

resumen (turnout.linear)

El resultado de nuestro comando de resumen es:

Llamada:

glm (fórmula = votado ~ ponderado a distancia, familia = binomi

al (enlace = “probit”),

datos = votación)

Residuos de desviación:

Mín. 1T Mediana 3T Máx.

-1,9732 0,5550 0,5550 0,5776 0,6644

Coeficientes:

Estimar Std. Error z valor Pr (> | z |)

(Intercepción) 1.068134 0.009293 114.942 <2e-16

ponderado por distancia -0,055074 0,011724 -4,698 2,63e-06

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05.

0,1 1

(Parámetro de dispersión para la familia binomial llevado a

ser 1)

Desviación nula: 37788 sobre 44896 grados de libertad

Desviación residual: 37766 sobre 44895 grados de libertad

AIC: 37770

Número de iteraciones de puntuación de Fisher: 4

El diseño de estos resultados del modelo probit es similar a los resultados del modelo logit. Sin embargo, tenga en cuenta que cambiar la distribución del término de error latente a una distribución normal cambia la escala de los coeficientes, por lo que los valores serán diferentes entre los modelos logit y probit. Las implicaciones sustantivas suelen ser similares entre los modelos, por lo que el usuario debe decidir qué modelo funciona mejor en términos de suposiciones e interpretación de los datos disponibles.

7.1.3 Interpretación de resultados logit y probit Una característica importante de los GLM es que el uso de una función de enlace hace que los coeficientes sean más difíciles de interpretar. Con un modelo de regresión lineal, estimado en el Cap. 6 , podríamos simplemente interpretar el coeficiente en términos de cambio en el valor esperado del resultado mismo, manteniendo iguales las otras variables. Sin embargo, con un GLM, la media del resultado se ha transformado y el coeficiente habla del cambio en la media transformada. Por lo tanto, para análisis como modelos logit y probit, necesitamos tomar pasos adicionales para interpretar el efecto que tiene una entrada en el resultado de interés.

Para un modelo de regresión logística, el analista puede calcular rápidamente la razón de probabilidades para cada coeficiente simplemente tomando el exponencial del coeficiente. 6 Recuerde que las probabilidades de un evento son la razón entre la probabilidad de que ocurra el evento y la probabilidad de que no ocurra: pag1 - p . La razón de probabilidades nos dice el factor multiplicativo por el cual las probabilidades cambiarán para un aumento unitario en el predictor. Dentro de R , si queremos la razón de probabilidades para nuestro coeficiente de distancia en la tabla 7.1 , simplemente escribimos:

exp (incluyendo coeficientes de $ lineales [-1])

Esta sintaxis tomará el exponencial de cada estimación de coeficiente de nuestro modelo, sin importar el número de covariables. El [-1] omite la intersección, por lo que una razón de posibilidades no tendría sentido. Teniendo solo un predictor, la impresión en este caso es:

distanciainc

0.6097611

Debemos tener cuidado al interpretar el significado de las razones de probabilidades. En este caso, para un aumento de un punto en la distancia del partido en el poder en la escala de ideología, las probabilidades de que un encuestado vote por el partido en el poder se reducen en un factor de 0,61. (Con múltiples predictores, necesitaríamos agregar la advertencia ceteris paribus .) Si, en lugar de interpretarlo como un factor multiplicativo, el analista prefirió discutir el cambio en términos porcentuales, escriba:

100 \* (exp (con coeficientes de $ lineales [-1]) - 1)

En este caso, se devuelve un valor de -39.02389. Por lo tanto, podemos decir: para un aumento de un punto en la distancia del partido en el poder en la escala de ideología, las probabilidades de que un encuestado vote por el partido en el poder disminuyen en un 39%. Sin embargo, recuerde que todas estas declaraciones se relacionan específicamente con las probabilidades , por lo que en este caso nos referimos a una disminución del 39% en la relación entre la probabilidad de votar por el titular y la probabilidad de votar por cualquier otro partido.

Una interpretación alternativa que a menudo es más fácil de explicar en el texto es informar las probabilidades predichas a partir de un modelo. Para un modelo de regresión logística, ingresar las predicciones de la función lineal (las utilidades latentes) en la función de distribución acumulativa logística produce la probabilidad predicha de que el resultado tome un valor de 1. Un enfoque simple para ilustrar intuitivamente el efecto de un predictor es trazar las probabilidades predichas en cada valor que puede tomar un predictor, lo que muestra cómo cambia la probabilidad de forma no lineal a medida que cambia el predictor. Procedemos primero creando nuestras probabilidades predichas:

distancias <-seq (0,10, por = .1)

entradas <-cbind (1, distancias)

colnames (entradas) <- c (“constante,” “distanciainc”)

entradas <-as.data.frame (entradas)

Forecast.linear <-predict (incluido lineal, nuevos datos = entradas,

type = "respuesta")

En la primera línea, creamos una ss influencia de posibles distancias desde el partido en el poder, que van desde el mínimo (0) al máximo (10) en incrementos pequeños (0,1). Luego creamos una matriz denominada entradas que almacena los valores de predictores de interés para todos los predictores en nuestro modelo (usando el comando c olumn bind , cbind , para combinar dos vectores como columnas en una matriz). Posteriormente, nombramos las columnas para que coincidan con nuestros nombres de variables y recategorizamos esta matriz como un marco de datos . En la línea final, usamos el comando de predicción , que guarda las probabilidades predichas en un vector. Observe el uso de newdataopción para especificar nuestro marco de datos de valores predictores y la opción de tipo para especificar que queremos nuestros valores predichos en la escala de respuesta . Al establecer esto en la escala de respuesta, el comando devuelve probabilidades pronosticadas de votar por el partido en el poder en cada distancia hipotética.

Como alternativa al modelo en el que votar por el titular es una función de la distancia ideológica lineal entre el votante y el partido, también ajustamos un modelo utilizando la distancia al cuadrado. Podemos calcular fácilmente la probabilidad predicha de este modelo alternativo contra el valor de la distancia en su escala original. Nuevamente, las probabilidades predichas se calculan escribiendo:

entradas2 <-cbind (1, distancias ^ 2)

colnames (entradas2) <- c (“constante,” “distanciaincsq”)

input2 <-as.data.frame (input2)

Forecast.squared <-predict (incluido cuadrado, newdata = input2,

type = "respuesta")

En este caso, usamos las distancias vectoriales originales que capturaron valores predictores hipotéticos y los cuadramos. Al usar estos valores al cuadrado, guardamos nuestras probabilidades predichas del modelo alternativo en el vector de pronóstico .

Para graficar las probabilidades predichas de cada modelo en el mismo espacio, escribimos:

plot (y = previsión.lineal, x = distancias, ylim = c (0, .6), tipo = “l,”

lwd = 2, xlab = "", ylab = "")

líneas (y = pronóstico.cuadrado, x = distancias, lty = 2, col = “azul,” lwd = 2)

leyenda (x = 6, y = .5, leyenda = c (“lineal,” “cuadrado”), lty = c (1,2),

col = c ("negro", "azul"), lwd = 2)

mtext (“Distancia ideológica,” lado = 1, línea = 2.75, cex = 1.2)

mtext (“Probabilidad de votar por el titular,” lado = 2,

línea = 2.5, cex = 1.2)

En la primera línea, graficamos las probabilidades predichas del modelo con distancia lineal. En el eje vertical ( y ) están las probabilidades y en el eje horizontal ( x ) están los valores de la distancia. Limitamos las probabilidades entre 0 y 0,6 para ver más de cerca los cambios, establecemos type = “l” para producir una gráfica de línea y usamos la opción lwd = 2 para aumentar el grosor de la línea. También configuramos las etiquetas de los ejes X e Y para que estén vacías ( xlab = "“, ylab =”" ) para que podamos completar las etiquetas más tarde con un comando más preciso. En la segunda línea, agregamos otra línea a la figura abierta de las probabilidades predichas del modelo con distancia al cuadrado. Esta vez, coloreamos la línea de azul y la hacemos discontinua ( lty = 2) para distinguirlo de las probabilidades predichas del otro modelo. En la tercera línea, agregamos una leyenda a la gráfica, ubicada en las coordenadas donde x = 6 e y = 0.5 , que distingue las líneas en función de las distancias lineales y cuadradas. Finalmente, en las dos últimas líneas agregamos etiquetas de eje usando el comando mtext : La opción lateral nos permite declarar en qué eje estamos escribiendo, el comando de línea determina qué tan lejos del eje se imprime la etiqueta y el comando cex nos permite para ampliar el tamaño de la fuente (al 120% en este caso). Los resultados completos se presentan en la Fig. 7.1 . . Como muestra la figura, el modelo con distancia al cuadrado responde mejor en valores medios, con una respuesta más plana en los extremos. Por lo tanto, Singh’s (2014a) La conclusión de que la distancia lineal se ajusta mejor tiene implicaciones sustanciales para el comportamiento de los votantes. Abrir imagen en nueva ventanaFigura 7.1 Figura 7.1 Probabilidad pronosticada de votar por el partido en el poder en función de la distancia ideológica de los titulares, basada en una forma funcional lineal y cuadrática

Como ejemplo final de cómo informar las probabilidades predichas, recurrimos a un ejemplo del modelo probit que estimamos de participación. Las probabilidades pronosticadas se calculan de manera similar para los modelos probit, excepto que las predicciones lineales (o utilidades) ahora se ingresan en una función de distribución acumulativa normal. En este ejemplo, agregaremos a nuestra presentación de probabilidades predichas al incluir intervalos de confianza alrededor de nuestras predicciones, que transmiten al lector el nivel de incertidumbre en nuestro pronóstico. Comenzamos como lo hicimos en el último ejemplo, creando un marco de datos de valores de datos hipotéticos y produciendo probabilidades predichas con ellos:

wght.dist <-seq (0,4, por = .1)

entradas.3 <-cbind (1, wght.dist)

colnames (input.3) <- c (“constante,” “ponderado por distancia”)

input.3 <-as.data.frame (input.3)

Forecast.probit <-predict (turnout.linear, newdata = inputs.3,

type = "link", se.fit = TRUE)

En este caso, la distancia ideológica ponderada del partido ideológico más cercano es nuestro único predictor. Este predictor varía de aproximadamente 0 a 4, por lo que creamos un vector que abarca esos valores. En la última línea del código anterior, hemos cambiado dos características: Primero, hemos especificado type = “link” . Esto significa que nuestras predicciones son ahora predicciones lineales de la utilidad latente, y no las probabilidades en las que estamos interesados. (Esto se corregirá en un momento). En segundo lugar, hemos agregado la opción se.fit = TRUE , que nos proporciona un error estándar de cada predicción lineal. Nuestro objeto de salida, forecast.probit ahora contiene tanto los pronósticos lineales como los errores estándar.

La razón por la que guardamos las utilidades lineales en lugar de las probabilidades es que al hacerlo nos será más fácil calcular los intervalos de confianza que se mantienen dentro de los límites de probabilidad de 0 y 1. Para hacer esto, primero calculamos los intervalos de confianza de las predicciones lineales. . Para el nivel de confianza del 95%, escribimos:

lower.ci <-forecast.probit $ fit-1.95996399 \* forecast.probit $ se.fit

upper.ci <-forecast.probit $ fit + 1.95996399 \* forecast.probit $ se.fit

Observe que al llamar a Forecast.probit $ fit obtenemos las predicciones lineales de servicios públicos, y al llamar a Forecast.probit $ se.fit llamamos a los errores estándar de nuestro pronóstico. 1.95996399 es el valor crítico del 95% de dos colas de una distribución normal. Ahora que tenemos los vectores de los límites inferior y superior del intervalo de confianza, podemos insertarlos en la función de distribución acumulativa normal para poner los límites en la escala de probabilidad predicha.

Ahora podemos graficar las probabilidades predichas con intervalos de confianza de la siguiente manera:

plot (y = pnorm (pronóstico.probit $ ajuste), x = wght.dist, ylim = c (.7, .9),

type = "l", lwd = 2, xlab = "Distancia ideológica ponderada",  
  
 ylab = "Probabilidad de participación")

líneas (y = pnorm (lower.ci), x = wght.dist, lty = 3, col = “red,” lwd = 2)

líneas (y = pnorm (upper.ci), x = wght.dist, lty = 3, col = “red,” lwd = 2)

En la primera línea, trazamos las propias probabilidades predichas. Para obtener las probabilidades para el eje vertical, escribimos y = pnorm (pronóstico.probit $ ajuste) . La función pnorm es la función de distribución acumulativa normal, por lo que convierte nuestras predicciones de utilidad lineal en probabilidades reales. Mientras tanto, x = wght.dist coloca los posibles valores de la distancia ponderada a la parte más cercana en el eje horizontal. En la segunda línea, graficamos el límite inferior del intervalo de confianza del 95% de las probabilidades predichas. Aquí, pnorm (lower.ci) convierte el pronóstico del intervalo de confianza en la escala de probabilidad. Finalmente, repetimos el proceso en la línea tres para graficar el límite superior del intervalo de confianza. La salida completa se puede ver en la Fig. 7.2 . Una característica notable de este gráfico es que el intervalo de confianza se vuelve notablemente amplio para los valores más grandes de la distancia ponderada. Esto se debe a que la media de la variable es baja y hay pocas observaciones en estos valores más altos. Abrir imagen en nueva ventanaFigura 7.2 Figura 7.2 Probabilidad prevista de votar en función de la distancia ideológica ponderada del partido más cercano, con intervalos de confianza del 95%

Las probabilidades predichas en ambos casos fueron simples porque incluían solo un predictor. Para cualquier GLM, incluido un modelo logit o probit, las probabilidades predichas y su nivel de respuesta dependen del valor de todas las covariables. Siempre que un investigador tenga múltiples predictores en un modelo GLM, los valores razonables de las variables de control deben incluirse en los pronósticos. Ver secc. 7.3.3 para ver un ejemplo del uso de la función de predicción para un GLM que incluye múltiples predictores.

7.2 Resultados ordinales Pasamos ahora a las medidas de resultado ordinales. Las variables ordinales tienen múltiples categorías como respuestas que se pueden clasificar de menor a mayor, pero aparte de las clasificaciones, los valores numéricos no tienen un significado inherente. Como ejemplo de una variable de resultado ordinal, nuevamente usamos datos de encuestas del CSES, esta vez de Singh (2014b) estudio de satisfacción con la democracia. En relación con el ejemplo anterior, estos datos tienen un alcance más amplio, incluidos 66.908 encuestados de 62 elecciones. Las variables de este conjunto de datos son las siguientes: satisfacción: Nivel de satisfacción de la encuestada con la democracia. Escala ordinal codificada 1 (nada satisfecho), 2 (poco satisfecho), 3 (bastante satisfecho) o 4 (muy satisfecho).

cntryyear: Una variable de carácter que enumera el país y el año de la elección.

cntryyearnum: Un índice numérico que identifica el país y el año de la elección.

libertad: Freedom House puntúa el nivel de libertad de un país. Las puntuaciones van de -5,5 (menos gratis) a -1 (más gratis).

El crecimiento del PIB: Crecimiento porcentual del Producto Interno Bruto (PIB).

gdppercapPPP: PIB per cápita, calculado utilizando paridad de precios de compra (PPA), encadenado a 2000 dólares internacionales, en miles de dólares.

IPC: Indice de Percepción de la corrupción. Las puntuaciones van de 0 (menos corrupto) a 7,6 (más corrupto).

eficacia: La Demandada cree que votar puede marcar la diferencia. Escala ordinal de 1 (en desacuerdo) a 5 (de acuerdo).

educ: Indicador codificado 1 si el encuestado se graduó de la universidad, 0 si no.

se abstuvo: Indicador codificado 1 si el encuestado se abstuvo de votar, 0 si el encuestado votó.

prez: Indicador codificado 1 si el país tiene un sistema presidencial, 0 en caso contrario.

majoritarian\_prez: Indicador codificado 1 si el país tiene un sistema mayoritario, 0 en caso contrario.

ganador: Indicador codificado 1 si el encuestado votó por el partido ganador, 0 si no.

vote\_ID: Indicador codificado 1 si el encuestado votó por el partido con el que se identifica, 0 en caso contrario.

vote\_affect: Indicador codificado con 1 si el encuestado votó por el partido al que calificó más alto, 0 en caso contrario.

vote\_ideo: Indicador codificado 1 si el encuestado votó por el partido más similar en ideología, 0 si no.

Optimismo: Escala de optimización de votos que va de 0 a 3, codificada agregando vote\_ID , vote\_affect y vote\_ideo .

ganadorXvoted\_ID: Término de interacción entre votar por el ganador y votar por identificación de partido.

ganadorXvoted\_affect: Término de interacción entre votar por el ganador y votar por el partido mejor calificado.

ganadorXvoted\_ideo: Término de interacción entre votar por el ganador y votar por similitud ideológica.

ganadorXoptimality: Término de interacción entre la votación por el ganador y la escala de optimización de votos.

Estos datos también están en formato Stata, por lo que si la biblioteca externa aún no está cargada, será necesario llamarla. Para cargar nuestros datos, descargue el archivo SinghEJPR.dta del Dataverse vinculado en la página vii o el contenido del capítulo vinculado en la página 97. Luego escriba:

biblioteca (extranjera)

satisfacción <-read.dta (“SinghEJPR.dta”)

En este ejemplo, deseamos modelar el nivel de satisfacción de cada encuestado con la democracia. Esta variable toma los valores 1, 2, 3 y 4, y solo podemos hacer declaraciones sobre qué valores reflejan mayores niveles de satisfacción. En otras palabras, podemos decir que un valor de 4 (“muy satisfecho”) refleja una mayor satisfacción que un valor de 3 (“bastante satisfecho”), un valor de 3 es más de 2 (“no muy satisfecho”), y por lo tanto, un valor de 4 es mayor que un valor de 2. Sin embargo, no podemos decir cuántomás satisfacción que refleja un valor en relación con otro, ya que los números no tienen un significado inherente más que proporcionar un orden de satisfacción. Para hacerlo, tendríamos que cuantificar adjetivos como “muy” y “bastante.” Por lo tanto, un modelo logit ordenado o probit ordenado será apropiado para este análisis.

Como nuestro primer ejemplo, Singh (2014b, Tabla SM2) se ajusta a un modelo en el que la satisfacción en democracia se basa en si el encuestado votó por el candidato más próximo ideológicamente, si el candidato votó por el ganador y la interacción entre estas dos variables. 7 El más importante de estos términos es la interacción, ya que pone a prueba la hipótesis de que los individuos que estuvieron del lado ganador y votaron por el partido más similar ideológicamente expresarán la mayor satisfacción.

En cuanto a los detalles, para los modelos de regresión ordinal que realmente hay que utilizar el comando especial polr (abreviatura de p roportional o DDS l ogistic r egression), que es parte de la MASA paquete. La mayoría de las distribuciones de R instalan automáticamente MASS , aunque todavía necesitamos cargarlo con el comando de la biblioteca . 8 Para cargar el paquete MASS y luego estimar un modelo logit ordenado con estos datos, teclearíamos:

biblioteca (MASA)

satisfacción $ satisfacción <-ordenados (as.factor (

satisfacción $ satisfacción))

ideol.satisfaction <-polr (satisfacción ~ vote\_ideo \* ganador +

abstencion + educ + eficacia + majoritarian\_prez +  
  
 freedom + gdppercapPPP + gdpgrowth + CPI + prez,  
  
 método = "logística", datos = satisfacción)

resumen (ideol. satisfacción)

Observe que recodificamos nuestra variable dependiente, satisfacción , usando el comando as.factor que se introdujo en la Sect. 2.4.2 Además, incorporamos esto dentro del comando ordenado para transmitir que el factor se puede ordenar numéricamente. Tuvimos que recodificar de esta manera porque el comando modelo polr requiere que el resultado se guarde como un vector de factor de clase . Mientras tanto, el lado derecho del modelo se parece a todos los demás modelos que hemos estimado hasta ahora separando los nombres de las variables con signos más. Tenga en cuenta que usamos notación interactiva con vote\_ideo \* ganador , para incluir ambos términos más el producto. 9 En conjunto, hemos modelado la satisfacción en función de si el votante votó por el partido ideológicamente más similar, votó por el ganador, un término de interacción entre los dos y varias otras variables de control a nivel individual y nacional. Dentro del comando, la opción method = “logistic” especifica que deseamos estimar un modelo logit ordenado en lugar de usar otra función de enlace. Al final de la línea, especificamos nuestra opción de datos como de costumbre para apuntar a nuestro conjunto de datos de interés.

Después de estimar este modelo, escribimos resumen (ideol.satisfaction) en la consola. La salida se ve así:

Llamada:

polr (fórmula = satisfacción ~ video\_ votada \* ganador + abstinencia +

educ + eficacia + majoritarian\_prez + freedom + gdppercap  
  
 PPP +  
  
gdpgrowth + CPI + prez, data = satisfacción, método =  
  
 "logístico")

Coeficientes:

Valor Std. Valor t de error

vote\_ideo -0.02170 0.023596 -0.9198

ganador 0.21813 0.020638 10.5694

abstinencia -0,25425 0,020868 -12,1838

educ 0,08238 0,020180 4,0824

eficacia 0,16246 0,006211 26,1569

majoritarian\_prez 0.05705 0.018049 3.1609

libertad 0.04770 0.014087 3.3863

gdppercapPPP 0.01975 0.001385 14.2578

gdpgcrecimiento 0,06653 0,003188 20,8673

IPC -0,23153 0,005810 -39,8537

prez -0.11503 0.026185 -4.3930

vote\_ideo: ganador 0.19004 0.037294 5.0957

Intercepciones:

Valor Std. Valor t de error

1 | 2 -2.0501 0.0584 -35.1284

2 | 3 -0,0588 0,0575 -1,0228

3 | 4 2,7315 0,0586 46,6423

Desviación residual: 146397,33

AIC: 146427.33

La salida muestra la estimación de cada coeficiente, el error estándar y el valor z . (Aunque R lo llama un valor t , los métodos de máxima verosimilitud suelen exigir relaciones z , como se mencionó anteriormente). Después de la presentación del coeficiente, se presentan tres puntos de corte bajo la etiqueta de intersecciones. Estos puntos de corte identifican el modelo al encontrar en qué lugar de una escala de utilidad latente se encuentra el límite entre que el encuestado elija 1 frente a 2 como respuesta, 2 frente a 3 y 3 frente a 4. Aunque los puntos de corte a menudo no son de interés sustancial, son importantes por el bien de pronosticar los resultados previstos. Las estadísticas de ajuste predeterminadas en la salida son la desviación residual y el AIC.

Los resultados se presentan de manera más formal en la Tabla 7.2 . Aunque la salida base de R omite el valor p , el usuario puede extraer fácilmente inferencias basadas en la información disponible: ya sea calculando intervalos de confianza, comparando el valor z con un valor crítico, o calculando los valores p él mismo. Por ejemplo, la hipótesis clave aquí es que el término de interacción sea positivo. Por lo tanto, podemos obtener nuestro valor p de una cola escribiendo: Cuadro 7.2 Modelo logit ordenado de satisfacción con la democracia, 62 elecciones transnacionales

Vaticinador

Estimar

Std. error

valor z

Pr (> |  z  |)

Votado por el partido próximo

-0.0217

0.0236

-0,9198

0.3577

Votado por el ganador

0.2181

0.0206

10.5694

0,0000

Votado próximo × ganador

0,1900

0.0373

5.0957

0,0000

Se abstuvo

-0,2542

0.0209

-12.1838

0,0000

Graduado de la Universidad

0.0824

0.0202

4.0824

0,0000

Eficacia

0.1625

0,0062

26.1569

0,0000

Sistema mayoritario

0.0571

0.0180

3.1609

0,0016

Libertad

0.0477

0.0141

3.3863

0,0007

Desarrollo economico

0.0197

0,0014

14.2578

0,0000

Crecimiento económico

0.0665

0,0032

20.8673

0,0000

Corrupción

-0,2315

0,0058

-39.8537

0,0000

Sistema presidencial

-0,1150

0.0262

-4,3930

0,0000

τ 1

-2.0501

0.0584

-35.1284

0,0000

τ 2

-0.0588

0.0575

-1.0228

0.3064

τ 3

2.7315

0.0586

46.6423

0,0000

Notas : N  = 66, 908. AIC = 146, 427. Datos de Singh (2014b)

1 pnorm (5.0957)

R imprime 1.737275e-07 , que en notación científica significa p  = 0. 00000017. Por lo tanto, concluiremos con un 99,9% de confianza de que el coeficiente del término de interacción es perceptiblemente mayor que cero. En la Tabla 7.2 , hemos optado por informar los valores p de dos colas . 10

Una característica interesante del uso de la función de enlace logit es que los resultados se pueden interpretar en términos de razones de probabilidades. Sin embargo, las razones de probabilidad deben calcularse e interpretarse de manera un poco diferente para un modelo ordinal. Ver largo (1997, págs. 138-140) para obtener una explicación completa. Para modelos logit ordenados, debemos exponenciar el valor negativo de un coeficiente e interpretar las probabilidades de estar en grupos más bajos en relación con los grupos más altos. A modo de ejemplo, las razones de probabilidades para nuestros coeficientes de la tabla 7.2 , junto con los cambios porcentuales en las probabilidades, se pueden generar de una vez:

exp (coeficientes de $ de satisfacción -ideol)

100 \* (exp (-ideol.satisfacción $ coeficientes) -1)

La impresión resultante de la segunda línea es:

vote\_ideo el ganador se abstuvo  
  
 2.194186 -19.597657 28.949139  
  
 educ eficacia majoritarian\_prez  
  
 -7.908003 -14.994659 -5.545347  
  
 libertad gdppercapPPP gdpgrowth  
  
 -4.658313 -1.955471 -6.436961  
  
 CPI prez vot\_ideo: ganador  
  
 26.053177 12.190773 -17.307376

Si quisiéramos interpretar el efecto de la eficacia, entonces, podríamos decir que para un aumento de un punto en una escala de eficacia de cinco puntos, las probabilidades de que un encuestado informe que no está “en absoluto satisfecho” con la democracia en relación con cualquiera de los tres categorías superiores disminuyen en un 15%, ceteris paribus. Además, las probabilidades de que un encuestado informe “nada satisfecho” o “no muy satisfecho” en relación con las dos categorías superiores también disminuyen en un 15%, todo lo demás igual. Además, las probabilidades de que un encuestado informe uno de los tres niveles de satisfacción inferiores en relación con la categoría más alta de “muy satisfecho” disminuyen en un 15%, manteniendo constantes los demás predictores. En general, entonces, podemos interpretar una razón de probabilidades para un logit ordenado como la configuración de las probabilidades de todas las opciones por debajo de un umbral en relación con todas las opciones por encima de un umbral.

Como segundo ejemplo, pasamos ahora a un modelo de satisfacción del votante que se centra no en el papel de la proximidad ideológica en la elección del voto, sino en qué partido evaluó mejor el votante cuando se le pidió que calificara a los partidos. Nuevamente, la interacción entre votar por el partido mejor calificado y también votar por el partido ganador es el principal coeficiente de interés. Esta vez, también probaremos una función de enlace diferente y estimaremos un modelo probit ordenado en lugar de un modelo logit ordenado. En este caso teclearemos:

fect.satisfaction <-polr (satisfacción ~ vote\_affect \* ganador +

abstencion + educ + eficacia + majoritarian\_prez +  
  
 freedom + gdppercapPPP + gdpgrowth + CPI + prez,  
  
 método = "probit", datos = satisfacción)

Además de cambiar una de las variables interactuadas, la única diferencia entre este código y el comando anterior para el comando logit ordenado es la especificación de method = “probit” . Esto cambia un poco la escala de nuestros coeficientes, pero las implicaciones sustantivas de los resultados son generalmente similares independientemente de esta elección. Al escribir resumen (afecta la satisfacción) , obtenemos el resultado:

Llamada:

polr (fórmula = satisfacción ~ efecto\_ votada \* ganador +

abstenido +

educ + eficacia + majoritarian\_prez + freedom +  
  
 gdppercapPPP +  
  
gdpgrowth + CPI + prez, data = satisfacción, método  
  
 = "probit")

Coeficientes:

Valor Std. Valor t de error

vote\_affect 0.03543 0.0158421 2.237

ganador 0.04531 0.0245471 1.846

abstinencia -0,11307 0,0170080 -6,648

educ 0,05168 0,0115189 4,487

eficacia 0.09014 0.0035177 25.625

majoritarian\_prez 0.03359 0.0101787 3.300

libertad 0.03648 0.0082013 4.448

gdppercapPPP 0.01071 0.0007906 13.546

gdpgcrecimiento 0.04007 0.0018376 21.803

IPC -0,12897 0,0033005 -39,075

prez -0.03751 0.0147650 -2.540

vote\_affect: ganador 0.14278 0.0267728 5.333

Intercepciones:

Valor Std. Valor t de error

1 | 2 -1,1559 0,0342 -33,7515

2 | 3 -0,0326 0,0340 -0,9586

3 | 4 1,6041 0,0344 46,6565

Desviación residual: 146698.25

AIC: 146728.25

Una vez más, nuestra hipótesis de interés está respaldada por los resultados, con un efecto positivo y perceptible en la interacción entre votar por el partido ganador y votar por el partido mejor calificado.

7.3 Recuentos de eventos Como tercer tipo de GLM, recurrimos a modelos de recuento de eventos. Siempre que nuestra variable dependiente sea el número de eventos que ocurren dentro de un período de tiempo definido, la variable tendrá la característica de que nunca puede ser negativa y debe tomar un valor discreto (por ejemplo, 0,1,2,3,4, …). Por lo tanto, los resultados del recuento tienden a tener un fuerte sesgo hacia la derecha y una distribución de probabilidad discreta como la distribución binomial negativa o de Poisson.

Como ejemplo de datos de recuento, ahora volvemos a Peake y Eshbaugh-Soha (2008) datos que se discutieron previamente en el Cap. 3 Recuerde que la variable de resultado en este caso es el número de noticias de televisión relacionadas con la política energética en un mes determinado. (Consulte el Capítulo  3 para obtener detalles adicionales sobre los datos). El número de noticias en un mes ciertamente es un recuento de eventos. Sin embargo, tenga en cuenta que debido a que estos son datos mensuales, dependen del tiempo , que es una característica que ignoramos en este momento. En el Cap. 9 revisamos este tema y consideramos modelos que dan cuenta del tiempo. Por ahora, sin embargo, esto ilustra cómo nosotros, los modelos de recuento suelen encajar en R .

Primero, cargamos los datos nuevamente: 11

pres.energy <-read.csv (“PESenergy.csv”)

Después de ver las estadísticas descriptivas de nuestras variables y visualizar los datos como hicimos en el Cap. 3 , ahora podemos pasar a ajustar un modelo.

7.3.1 Regresión de Poisson El modelo de conteo más simple que podemos ajustar es un modelo de Poisson. Si tuviéramos que escribir:

energy.poisson <-glm (Energía ~ rmn1173 + grf0175 + grf575 + jec477 +

jec1177 + jec479 + embargo + rehenes + oilc + Aprobación + Desempleo,  
  
 family = poisson (link = log), data = pres.energy)

Esto se ajustará a un modelo de regresión de Poisson en el que la cobertura televisiva de la política energética es una función de seis mandatos para los discursos presidenciales, un indicador del embargo petrolero árabe, un indicador de la crisis de los rehenes en Irán, el precio del petróleo, la aprobación presidencial y la tasa de desempleo. 12 Observe que esta vez, establecemos family = poisson (link = log) , declarando la distribución del resultado como Poisson y especificando nuestra función log link. Si escribimos resumen (energy.poisson) en la consola, R devuelve el siguiente resultado:

Llamada:

glm (fórmula = Energía ~ rmn1173 + grf0175 + grf575 +

jec477 +

jec1177 + jec479 + embargo + rehenes + oilc +  
  
 Aprobación +  
  
Desempleo, familia = poisson (enlace = log), datos = pres.  
  
 energía)

Residuos de desviación:

Mín. 1T Mediana 3T Máx.

-8.383 -2.994 -1.054 1.536 11.399

Coeficientes:

Estimar Std. Error z valor Pr (> | z |)

(Intercepción) 13.250093 0.329121 40.259 <2e-16 \*\*\*

rmn1173 0,694714 0,077009 9,021 <2e-16 \*\*\*

grf0175 0.468294 0.096169 4.870 1.12e-06 \*\*\*

grf575 -0.130568 0.162191 -0.805 0.420806

jec477 1.108520 0.122211 9.071 <2e-16 \*\*\*

jec1177 0.576779 0.155511 3.709 0.000208 \*\*\*

jec479 1.076455 0.095066 11.323 <2e-16 \*\*\*

embargo 0.937796 0.051110 18.349 <2e-16 \*\*\*

rehenes -0.094507 0.046166 -2.047 0.040647 \*

aceitec -0,213498 0,008052 -26,515 <2e-16 \*\*\*

Aprobación -0.034096 0.001386 -24.599 <2e-16 \*\*\*

Desempleo -0.090204 0.009678 -9.321 <2e-16 \*\*\*

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05. 0,1 1

(Parámetro de dispersión para la familia de Poisson tomado como 1)

Desviación nula: 6009.0 en 179 grados de libertad

Desviación residual: 2598,8 en 168 grados de libertad

AIC: 3488,3

Número de iteraciones de puntuación de Fisher: 5

El formato de las estimaciones de coeficientes de salida, errores estándar, z , y p -En caso de ser muy familiar a estas alturas, como son la desviación y las puntuaciones de la AIC. En este caso, la función de enlace es simplemente un logaritmo, por lo que, aunque los coeficientes en sí mismos no son muy significativos, la interpretación sigue siendo simple. Como una opción, si tomamos el exponencial de un coeficiente, esto nos ofrece una relación de conteo , que nos permite deducir el cambio porcentual en el conteo esperado para un cambio en la variable de entrada. Por ejemplo, si quisiéramos interpretar el efecto de la aprobación presidencial, podríamos escribir:

exp (-. 034096)

Aquí, simplemente insertamos el coeficiente estimado de la salida impresa. El resultado nos da una relación de recuento de 0,9664787. Podríamos interpretar esto como un significado para un aumento de un punto porcentual en el índice de aprobación del presidente, la cobertura de la política energética disminuye en un 3.4% en promedio y manteniendo todos los demás predictores iguales. Como una forma rápida de obtener la relación de recuento y el cambio porcentual de cada coeficiente, podríamos escribir:

exp (coeficientes de energía.poisson $ [-1])

100 \* (exp (coeficientes de energía.poisson $ [-1]) - 1)

En ambas líneas, el índice [-1] para el vector de coeficientes descarta el término de intersección, para el cual no queremos una razón de recuento. La impresión de la segunda línea dice:

rmn1173 grf0175 grf575 jec477 jec1177

100.313518 59.726654 -12.240295 202.987027 78.029428

jec479 embargo rehenes oilc Aprobación

193.425875 155.434606 -9.017887 -19.224639 -3.352127

Desempleo

-8.625516

De esta lista, podemos simplemente leer los cambios porcentuales para un aumento de una unidad en la entrada, manteniendo iguales las otras entradas. Para obtener un medio gráfico de interpretación, consulte la secc. 7.3.3 .

7.3.2 Regresión binomial negativa Una característica intrigante de la distribución de Poisson es que la varianza es la misma que la media. Por lo tanto, cuando modelamos el logaritmo de la media, nuestro modelo modela simultáneamente la varianza. Sin embargo, a menudo encontramos que la varianza de nuestra variable de conteo es más amplia de lo que esperaríamos dadas las covariables, un fenómeno llamado sobredispersión . La regresión binomial negativa ofrece una solución a este problema al estimar un parámetro de dispersión adicional que permite que la varianza condicional difiera de la media condicional.

En R , n egative b modelos de regresión inomial realmente requieren un comando especial de la MASA biblioteca llamada glm.nb . Si la biblioteca MASS no está cargada, asegúrese de escribir biblioteca (MASS) primero. Luego, podemos ajustar el modelo binomial negativo escribiendo:

energy.nb <-glm.nb (Energía ~ rmn1173 + grf0175 + grf575 + jec477 +

jec1177 + jec479 + embargo + rehenes + oilc + Aprobación + Desempleo,  
  
 datos = pres.energy)

Observe que la sintaxis es similar a la del comando glm , pero no hay una opción de familia ya que el comando en sí lo especifica. Al escribir resumen (energy.nb), se imprimen los siguientes resultados:

Llamada:

glm.nb (fórmula = Energía ~ rmn1173 + grf0175 + grf575 +

jec477 +

jec1177 + jec479 + embargo + rehenes + oilc +  
  
 Aprobación +  
  
Desempleo, datos = pres.energy, init.theta =  
  
 2.149960724, enlace = registro)

Residuos de desviación:

Mín. 1T Mediana 3T Máx.

-2,7702 -0,9635 -0,2624 0,3569 2,2034

Coeficientes:

Estimar Std. Error z valor Pr (> | z |)

(Intercepción) 15.299318 1.291013 11.851 <2e-16 \*\*\*

rmn1173 0,722292 0,752005 0,960 0,33681

grf0175 0,288242 0,700429 0,412 0,68069

grf575 -0,227584 0,707969 -0,321 0,74786

jec477 0.965964 0.703611 1.373 0.16979

jec1177 0,573210 0,702534 0,816 0,41455

jec479 1,141528 0,694927 1,643 0,10045

embargo 1.140854 0.350077 3.259 0.00112 \*\*

rehenes 0.089438 0.197520 0.453 0.65069

aceitec -0,276592 0,030104 -9,188 <2e-16 \*\*\*

Aprobación -0.032082 0.005796 -5.536 3.1e-08 \*\*\*

Desempleo -0.077013 0.037630 -2.047 0.04070 \*

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05. 0,1 1

(Parámetro de dispersión para la familia Binomial negativa (2.15)

tomado

ser 1)

Desviación nula: 393.02 en 179 grados de libertad

Desviación residual: 194,74 en 168 grados de libertad

AIC: 1526,4

Número de iteraciones de puntuación de Fisher: 1

Theta: 2.150  
  
 Std. Err .: 0.242

2 x probabilidad logarítmica: -1500,427

Los coeficientes informados en este resultado se pueden interpretar de la misma manera que los coeficientes de un modelo de Poisson porque ambos modelan el logaritmo de la media. La adición clave, informada al final de la impresión, es el parámetro de dispersión θ . En este caso, nuestra estimación es θ^= 2,15 , y con un error estándar de 0,242, el resultado es discernible. Esto indica que la sobredispersión está presente en este modelo. De hecho, muchas de las inferencias extraídas varían entre los modelos de Poisson y binomial negativo. Los dos modelos se presentan uno al lado del otro en la Tabla 7.3 . Como muestran los resultados, muchos de los resultados discernibles del modelo de Poisson no son discernibles en el modelo binomial negativo. Además, el AIC es sustancialmente más bajo para el modelo binomial negativo, lo que indica un mejor ajuste incluso cuando se penaliza por el parámetro de sobredispersión adicional. Cuadro 7.3 Dos modelos de conteo de nuevas historias de televisión mensuales sobre política energética, 1969-1983

Poisson

Binomio negativo

Parámetro

Estimar

Std. error

Pr (> |  z  |)

Estimar

Std. error

Pr (> |  z  |)

Interceptar

13.2501

0.3291

0,0000

15.2993

1.2910

0,0000

Nixon 11/73

0,6947

0.0770

0,0000

0,7223

0,7520

0.3368

Ford 1/75

0,4683

0.0962

0,0000

0.2882

0,7004

0,6807

Ford 5/75

−0,1306

0.1622

0.4208

−0,2276

0,7080

0,7479

Carter 4/77

1.1085

0.1222

0,0000

0.9660

0,7036

0.1698

Carter 11/77

0.5768

0.1555

0,0002

0.5732

0,7025

0.4145

Carter 4/79

1.0765

0.0951

0,0000

1,1415

0,6949

0.1005

Embargo de petróleo árabe

0,9378

0.0511

0,0000

1.1409

0.3501

0,0011

Crisis de rehenes en Irán

−0,0945

0.0462

0.0406

0.0894

0,1975

0,6507

Precio del aceite

−0,2135

0,0081

0,0000

−0,2766

0.0301

0,0000

Aprobación presidencial

−0,0341

0,0014

0,0000

−0,0321

0,0058

0,0000

Desempleo

−0,0902

0,0097

0,0000

−0,0770

0.0376

0.0407

θ

2.1500

0.2419

0,0000

AIC

3488.2830

1526.4272

Notas : N  = 180. Datos de Peake y Eshbaugh-Soha (2008)

7.3.3 Trazado de recuentos previstos Si bien las razones de conteo son sin duda una forma sencilla de interpretar los coeficientes de los modelos de conteo, también tenemos la opción de graficar nuestros resultados. En este caso, modelamos el logaritmo de nuestro parámetro medio, por lo que debemos exponenciar nuestra predicción lineal para predecir el recuento esperado dadas nuestras covariables. Al igual que con los modelos logit y probit, para el recuento de los desenlaces del predicen comando permite que la previsión fácil.

Supongamos que quisiéramos graficar el efecto de la aprobación presidencial sobre el número de noticias de televisión sobre la energía, según los dos modelos de la tabla 7.3 . Esta situación contrasta un poco con los gráficos que creamos en la Secta. 7.1.3 . En todos los ejemplos logit y probit, solo teníamos un predictor. Por el contrario, en este caso tenemos varios otros predictores, por lo que tenemos que establecerlos en valores alternativos plausibles. Para este ejemplo, estableceremos el valor de todos los predictores de variables ficticias en su valor modal de cero, mientras que el precio del petróleo y el desempleo se establecen en su media. Si no insertamos valores razonables para las covariables, los recuentos predichos no se parecerán a la media real y el tamaño del efecto no será razonable. 13En este ejemplo, la forma en que usamos el predecir comando para recuentos promedio de pronóstico con múltiples predictores se puede utilizar exactamente de la misma manera para un logit o probit para pronosticar probabilidades predichas con varios predictores.

En cuanto a los detalles, en nuestros datos, la variable aprobar varía entre un 24% de aprobación y un 72,3%. Por lo tanto, construimos un vector que incluye el rango completo de aprobación, así como los valores plausibles de todos los demás predictores:

aprobación <-seq (24,72.3, por = .1)

insumos.4 <-cbind (1,0,0,0,0,0,0,0,0, mean (pres.energy $ oilc),

aprobación, media (energía pres. $ Desempleo))

colnames (input.4) <- c (“constante,” “rmn1173,” “grf0175,”

"grf575", "jec477", "jec1177", "jec479", "embargo", "rehenes",  
  
 "oilc", "Aprobación", "Desempleo")

input.4 <-como.data.frame (input.4)

La primera línea de arriba crea el vector de valores hipotéticos de nuestro predictor de interés. La segunda línea crea una matriz de valores de datos hipotéticos: establece las variables indicadoras en cero, las variables continuas en sus medias y la aprobación en su rango de valores hipotéticos. La tercera línea nombra las columnas de la matriz después de las variables de nuestro modelo. En la última línea, la matriz de valores predictores se convierte en un marco de datos.

Una vez que tenemos la trama de datos de predictores en su lugar, podemos utilizar el predicen comando para pronosticar los recuentos esperados para los modelos binomial de Poisson y negativos:

Forecast.poisson <-predict (energy.poisson, newdata = inputs.4,

type = "respuesta")

Forecast.nb <-predict (energy.nb, newdata = inputs.4, type = “response”)

Estas dos líneas solo difieren en el modelo del que extraen estimaciones de coeficientes para el pronóstico. 14 En ambos casos, especificamos type = “response” para obtener predicciones en la escala de conteo.

Para graficar nuestros pronósticos de cada modelo, podemos escribir:

plot (y = previsión.poisson, x = aprobación, tipo = “l,” lwd = 2,

ylim = c (0,60), xlab = "Aprobación presidencial",  
  
 ylab = "Recuento previsto de historias de política energética")

líneas (y = previsión.nb, x = aprobación, lty = 2, col = “azul,” lwd = 2)

leyenda (x = 50, y = 50, leyenda = c (“Poisson,” “Binomio negativo”),

lty = c (1,2), col = c ("negro", "azul"), lwd = 2)

La primera línea traza las predicciones de Poisson como una línea con la opción type = “l” . La segunda línea agrega las predicciones binomiales negativas, coloreando la línea de azul y punzándola con lty = 2 . Finalmente, el comando de leyenda nos permite distinguir rápidamente qué línea representa qué modelo. La salida completa se presenta en la Fig.  7.3 . Las predicciones de los dos modelos son similares y muestran un efecto negativo similar de aprobación. El modelo binomial negativo tiene un pronóstico ligeramente más bajo con valores bajos de aprobación y un efecto de aprobación ligeramente menor, de modo que los conteos previstos se superponen con valores altos de aprobación. Abrir imagen en nueva ventanaFigura 7.3 Figura 7.3 Recuento previsto de historias de política energética en las noticias de televisión en función de la aprobación presidencial, manteniendo los predictores continuos en su media y los predictores nominales en su modo. Predicciones basadas en Poisson y resultados del modelo binomial negativo

Después de los primeros siete capítulos de este volumen, los usuarios ahora deberían poder realizar la mayoría de las tareas básicas para las que está diseñado el software estadístico: administrar datos, calcular estadísticas simples y estimar modelos comunes. En los cuatro capítulos restantes de este libro, pasamos ahora a las características únicas de R que permiten la flexibilidad mayor usuario aplicar métodos avanzados con paquetes desarrollados por otros usuarios y herramientas para la programación en R .

7.4 Problemas de práctica 1. Regresión logística: cargue la biblioteca externa y descargue un subconjunto de Singh (2015) datos de encuestas transnacionales sobre participación de votantes, el archivo stdSingh.dta , disponible en Dataverse enumerado en la página vii o el contenido del capítulo enumerado en la página 97. La variable de resultado es si el encuestado votó ( votó ). Un predictor clave, con el que interactúan varias variables, es el grado en que un ciudadano está sujeto a las reglas de votación obligatorias. Esto se mide con una escala de cuán severas son las reglas del voto obligatorio ( severidad ). Se deben interactuar cinco predictores con la gravedad : edad ( edad ), conocimiento político ( polinfrel ), ingresos ( ingresos ), eficacia ( eficacia ) y partidismo ( ID de partido).). Se deben incluir cinco predictores más solo para efectos aditivos: magnitud del distrito ( dist\_magnitude ), número de partidos ( enep ), margen de victoria ( vicmarg\_dist ), sistema parlamentario ( parlamentario ) y PIB per cápita ( desarrollo ). Todas las variables predictoras se han estandarizado. un. Estime un modelo de regresión logística con estos datos, incluidos los cinco términos de interacción.

1. ¿Cuál es la razón de probabilidades para el número de partidos? ¿Cómo interpretaría este término?
2. Grafique el efecto del PIB per cápita sobre la probabilidad de salir. Mantenga todos los predictores distintos del desarrollo en su media. Sugerencia: construya sobre el código que comienza en la página 107. Si usó la notación de interacción de R (por ejemplo, si edad \* severidad es un término en el modelo), entonces cuando cree un nuevo conjunto de datos de valores predictores, solo necesita definir valores para las variables originales y no para los productos. En otras palabras, necesitaría una columna para la edad , la gravedad y cualquier otro predictor, pero no para la edad × gravedad .
3. Bonificación: grafique el efecto de la edad en la probabilidad de participación en tres circunstancias: Cuando la severidad de las reglas de votación obligatoria es mínima, media y máxima. Mantenga todos los demás predictores, además de la edad y la gravedad, en su media. Su resultado final debe mostrar tres líneas de probabilidad predichas diferentes.
4. Logit ordenado: Los problemas de práctica del Capítulo 2 introdujeron los de Hanmer y Kalkan ( 2013) subconjunto del Estudio Electoral Nacional Estadounidense de 2004. Si aún no tiene estos datos, el archivo hanmerKalkanANES.dta se puede descargar desde el Dataverse vinculado en la página vii o desde el contenido del capítulo vinculado en la página 97. Cargue la biblioteca externa y abra estos datos. (Nuevamente, asegúrese de especificar la opción convert.factors = F ). Considere dos variables de resultado: evaluaciones económicas retrospectivas ( retecon , tomando valores ordinales codificados −1, −0.5, 0, 0.5 y 1) y evaluación de George W El manejo de Bush de la guerra en Irak ( bushiraq , tomando valores ordinales codificados 0, 0.33, 0.67 y 1). Hay siete variables predictoras: partidismo en una escala de siete puntos ( partyid), ideología en una escala de siete puntos ( ideol7b ), un indicador de si el encuestado es blanco ( blanco ), un indicador de si el encuestado es mujer ( mujer ), la edad del encuestado ( edad ), el nivel de educación en un escala de siete puntos ( educ1\_7 ) e ingresos en una escala de 23 puntos ( ingresos ). un. Estime un modelo logístico ordenado de evaluaciones económicas retrospectivas en función de los siete predictores.
5. ¿Cuál es la razón de probabilidades del coeficiente para mujeres? ¿Cómo interpretaría este término?
6. Estime un modelo logístico ordenado de evaluación del manejo de Bush de la guerra en Irak en función de los siete predictores.
7. ¿Cuál es la razón de probabilidades en el coeficiente de la escala de partidismo de siete puntos? ¿Cómo interpretaría este término?
8. Bonificación: Los resultados de un modelo están sesgados si hay causalidad recíproca , lo que significa que una de las variables independientes no solo influye en la variable dependiente, sino que también la variable dependiente influye en la variable independiente. Suponga que le preocupa el sesgo de causalidad recíproca en el modelo de evaluaciones económicas retrospectivas. ¿Qué variable o variables independientes serían más sospechosas de esta crítica?
9. Modelo de conteo: en los problemas de práctica de los capítulos 3 y 4 , presentamos el de Peake y Eshbaugh-Soha ( 2008) análisis de la cobertura de la póliza de medicamentos. Si no tiene sus datos anteriores, descargue drugCoverage.csv del Dataverse vinculado en la página vii o el contenido del capítulo vinculado en la página 97. La variable de resultado es la cobertura de noticias de medicamentos ( drugsmedia ), y las cuatro entradas son un indicador de un discurso sobre drogas que pronunció Ronald Reagan en septiembre de 1986 ( rwr86 ), un indicador de un discurso que pronunció George HW Bush en septiembre de 1989 ( ghwb89 ), el índice de aprobación del presidente ( aprobación ) y la tasa de desempleo ( desempleo ). 15 un. Estime un modelo de regresión de Poisson de cobertura de políticas de medicamentos como una función de los cuatro predictores.
10. Estime un modelo de regresión binomial negativa de cobertura de pólizas de medicamentos en función de los cuatro predictores. Según los resultados de sus modelos, ¿qué modelo es más apropiado, Poisson o binomial negativo? ¿Por qué?
11. Calcule la proporción de recuento del predictor de aprobación presidencial para cada modelo. ¿Cómo interpretaría cada cantidad?
12. Grafique los recuentos previstos de cada modelo dependiendo del nivel de desempleo, que van desde el mínimo al máximo de los valores observados. Mantenga las dos variables del discurso presidencial en cero y mantenga la aprobación presidencial en su media. Con base en esta figura, ¿qué puede decir sobre el efecto del desempleo en cada modelo?

Notas al pie 1 . En este caso, las estimaciones de los coeficientes que obtenemos son similares a las reportadas por Singh (2014a). Sin embargo, nuestros errores estándar son más pequeños (y por lo tanto los valores de z y p son mayores) porque Singh agrupa los errores estándar. Ésta es una idea útil porque los encuestados están anidados dentro de las elecciones, aunque los modelos multinivel (que Singh también informa) también abordan este tema — ver Sect. 8.1

2 . Los usuarios de LaTeX pueden crear una tabla similar a esta rápidamente escribiendo: library (xtable); xtable (incluido lineal) .

3 . Una explicación de cómo se derivan las propiedades inferenciales de este modelo se puede encontrar en Eliason (1993, págs. 26-27).

4 . La desviación se calcula como -2 veces la relación registrada entre la probabilidad ajustada y la probabilidad saturada. Formalmente, - 2 registroL1L2 , donde L 1 es la probabilidad ajustada y L 2 es la probabilidad saturada. R informa dos cantidades: la desviación nula calcula esto para un modelo de solo intercepción que siempre predice el valor modal, y la desviación residual calcula esto para el modelo informado.

5 . Además, en entornos avanzados para los que necesitamos desarrollar una distribución multivariante para múltiples variables de resultado, es relativamente fácil trabajar con la distribución normal.

6 . Esto se debe a que la función de enlace logit es el logaritmo de las probabilidades del evento.

7 . Este y el siguiente ejemplo no replican exactamente los resultados originales, que también incluyen efectos aleatorios por país y año. Además, el siguiente ejemplo ilustra la regresión probit ordenada, en lugar del modelo logístico ordenado del artículo original. Ambos ejemplos se basan en modelos que se encuentran en el material de apoyo en línea del sitio web European Journal of Political Research .

8 . Si un usuario necesita instalar el paquete, install.packages (“MASS”) hará el trabajo.

9 . Una especificación equivalente habría sido incluir vote\_ideo + ganador + ganadorXvoted\_ideo como tres términos separados de los datos.

10 . Desafortunadamente, el comando xtable no produce tablas LaTeX listas para usar para los resultados de polr . Sin embargo, al crear una matriz con los resultados relevantes, los usuarios de LaTeX pueden producir una tabla más rápido que la codificación manual, aunque son necesarias algunas revisiones del producto final. Intente lo siguiente:

coef <-c (ideol.satisfacción $ coeficientes, ideol.satisfacción $ zeta)

se <-sqrt (diag (vcov (satisfacción ideol.)))

z <-coef / se

p <-2 \* (1-pnorm (abs (z)))

xtable (cbind (coef, se, z, p), digits = 4)

11 . La nota al pie debe decir: ”Para los usuarios que no tienen el archivo a mano del Capítulo 3 , descargue el archivo del Dataverse vinculado en la página vii o el contenido del capítulo vinculado en la página 97.

12 . Tenga en cuenta que los términos del discurso presidencial se codifican con 1 solo en el mes del discurso y 0 en todos los demás meses. Los términos del embargo de petróleo y la crisis de rehenes se codificaron con 1 mientras estos eventos estaban en curso y 0 en caso contrario.

13 . Además de este enfoque de hacer predicciones utilizando valores centrales de variables de control, Hanmer y Kalkan (2013) argumentan que es preferible pronosticar los resultados basados ​​en los valores observados de las variables de control en el conjunto de datos. Se anima a los lectores a consultar su artículo para obtener más consejos sobre este tema.

14 . Como nota al margen, al usar los comandos de álgebra matricial de R , que se describen más adelante en el Cap. 10 , el usuario puede calcular fácilmente los recuentos previstos con una sintaxis alternativa. Por ejemplo, para el modelo binomial negativo, podríamos haber escrito: Forecast.nb <-exp (as.matrix (inputs.4)% \*% energy.nb $ coefficients) .

15 . Al igual que en el ejemplo del capítulo, estos son datos de series de tiempo, por lo que los métodos del Cap. 9 son más apropiados.

Material suplementario 318886\_1\_En\_7\_MOESM1\_ESM.zip (1.6 mb) Dataverse (2,154 KB) Referencias Black D (1948) Sobre el fundamento de la toma de decisiones en grupo. J Polit Econ 56 (1): 23–34 CrossRefGoogle Académico Downs A (1957) Una teoría económica de la democracia. Harper and Row, Nueva York Google Académico Eliason SR (1993) Estimación de máxima verosimilitud: lógica y práctica. Sage, Thousand Oaks, CA Google Académico Gill J (2001) Modelos lineales generalizados: un enfoque unificado. Sage, Thousand Oaks, CA Google Académico Hanmer MJ, Kalkan KO (2013) Detrás de la curva: aclarando el mejor enfoque para calcular las probabilidades pronosticadas y los efectos marginales a partir de modelos de variables dependientes limitadas. Am J Polit Sci 57 (1): 263–277 CrossRefGoogle Académico Hotelling H (1929) Estabilidad en competencia. Econ J 39 (153): 41–57 CrossRefGoogle Académico King G (1989) Metodología política unificadora. Cambridge University Press, Nueva York Google Académico Long JS (1997) Modelos de regresión para variables dependientes categóricas y limitadas. Sage, Thousand Oaks, CA zbMATHGoogle Académico Peake JS, Eshbaugh-Soha M (2008) El impacto en el establecimiento de la agenda de los principales discursos televisivos presidenciales. Polit Commun 25: 113-137 CrossRefGoogle Académico Singh SP (2014a) Funciones de pérdida de utilidad lineales y cuadráticas en la investigación del comportamiento electoral. J Theor Polit 26 (1): 35–58 CrossRefGoogle Académico Singh SP (2014b) No todos los ganadores de las elecciones son iguales: satisfacción con la democracia y la naturaleza del voto. Eur J Polit Res 53 (2): 308–327 CrossRefGoogle Académico Singh SP (2015) Voto obligatorio y cálculo de decisiones de participación. Polit Stud 63 (3): 548–568 CrossRefGoogle Académico

# 8 Uso de paquetes para aplicar modelos avanzados

Palabras clave: - Cadena Markov Monte Carlo - Muestra emparejada - Muestra de Monte Carlo de la cadena Markov - Llamada de rol - Partido titular

En los primeros siete capítulos de este libro, hemos tratado a R como un programa de software estadístico tradicional y hemos revisado cómo puede realizar la gestión de datos, informar estadísticas simples y estimar una variedad de modelos de regresión. En el resto de este libro, sin embargo, nos centraremos en la flexibilidad adicional que ofrece R , tanto en términos de capacidad de programación que está disponible para el usuario como en el suministro de herramientas aplicadas adicionales a través de paquetes . En este capítulo, nos enfocamos en cómo cargar lotes adicionales de código de paquetes escritos por el usuario puede agregar funcionalidad que muchos programas de software no permitirán. Aunque hemos utilizado paquetes para una variedad de propósitos en los siete capítulos anteriores (incluidos car , gmodels, y lattice , por nombrar algunos), aquí destacaremos los paquetes que habilitan métodos únicos. Si bien el sitio web de CRAN enumera numerosos paquetes que los usuarios pueden instalar en un momento dado, nos centraremos en cuatro paquetes particulares para ilustrar los tipos de funcionalidad que se pueden agregar.

El primer paquete que discutiremos, lme4 , permite a los usuarios estimar modelos multinivel, ofreciendo así una extensión a los modelos de regresión discutidos en los Capítulos. 6 y 7 (Bates et al. 2014). Los otros tres fueron desarrollados específicamente por científicos políticos para abordar los problemas de análisis de datos que encontraron en su investigación: MCMCpack permite a los usuarios estimar una variedad de modelos en un marco bayesiano utilizando la cadena de Markov Monte Carlo (MCMC) (Martin et al. 2011). cem permite al usuario realizar un emparejamiento exacto aproximado, un método para la inferencia causal con datos de campo (Iacus et al. 2009, 2011). Por último, wnominate permite al usuario escalar los datos de elección, como los datos de la lista legislativa, para estimar los puntos ideales ideológicos de los legisladores o encuestados (Poole y Rosenthal 1997; Poole y col. 2011). Las siguientes cuatro secciones considerarán cada paquete por separado, por lo que cada sección presentará su ejemplo de datos a su vez. Estas secciones están diseñadas para ofrecer una breve descripción de los tipos de capacidades que ofrecen los paquetes R , aunque algunos lectores pueden no estar familiarizados con los antecedentes de algunos de estos métodos. Se anima al lector interesado a consultar algunos de los recursos citados para aprender más sobre la teoría detrás de estos métodos.

8.1 Modelos multinivel con lme4 Habiendo discutido los modelos lineales en el Cap. 6 y varios ejemplos de modelos lineales generalizados en el Cap. 7 , pasamos ahora a una extensión de este tipo de modelos: modelos multinivel. Los modelos multinivel, o modelos jerárquicos, son apropiados siempre que los datos de interés tengan una estructura anidada o longitudinal. Una estructura anidada ocurre cuando se puede pensar que las observaciones están dentro o como parte de una unidad de nivel superior: un ejemplo de política común es estudiar los resultados del aprendizaje de los estudiantes, pero los estudiantes están anidados dentro de las aulas. En tal caso, el investigador debería tener en cuenta el hecho de que los estudiantes de la muestra no son independientes entre sí, pero es probable que sean similares si se encuentran en la misma clase. De manera similar, siempre que un investigador estudia individuos que han repetido observaciones a lo largo del tiempo, es razonable pensar que las observaciones referenciadas en el tiempo están integradas dentro de las observaciones del individuo. Por ejemplo,1986) datos introducidos por primera vez en el Cap. 4 , los ingresos de los participantes en su estudio se observan en 1974, 1975 y 1978. Algunos análisis de políticas pueden optar por considerar las tres observaciones temporales para cada individuo como anidadas dentro del caso de cada individuo. 1 Se pueden encontrar explicaciones más completas de los modelos multinivel en Scott et al. ( 2013) y Gelman y Hill (2007). Continuamos ampliando dos de nuestros ejemplos anteriores para ilustrar un modelo lineal multinivel y un modelo logit multinivel.

8.1.1 Regresión lineal multinivel En este ejemplo, volvemos a nuestro ejemplo del Cap. 6 sobre el número de horas que los docentes dedican a la docencia en el aula. Originalmente, ajustamos un modelo lineal utilizando mínimos cuadrados ordinarios (MCO) como nuestro estimador. Sin embargo, Berkman y Plutzer ( 2010) señalan que es probable que los profesores del mismo estado compartan características similares. Estas características podrían ser similitudes en la capacitación, en la cultura local o en la ley estatal. Para dar cuenta de estas similitudes no observadas, podemos pensar en los maestros como anidados dentro de los estados. Por esta razón, agregaremos un efecto aleatorio para cada estado. Los efectos aleatorios explican la correlación intraclase o la correlación de errores entre observaciones dentro del mismo grupo. En presencia de correlación intraclase, las estimaciones de MCO son ineficientes porque los términos de perturbación no son independientes, por lo que un modelo de efectos aleatorios da cuenta de este problema.

Primero, recargamos los datos de la Encuesta Nacional de Maestros de Biología de Escuelas Secundarias de la siguiente manera: 2

rm (lista = ls ())

biblioteca (extranjera)

evolución <-read.dta (“BPchap7.dta”)

evolución $ mujer [evolución $ mujer == 9] <- NA

evolución <-subconjunto (evolución,! is.na (mujer))

Recuerde que teníamos un puñado de observaciones de mujeres que debían ser recodificadas como perdidas. Como antes, subconjuntamos nuestros datos para omitir estas observaciones faltantes.

Para adaptarse a un modelo multinivel, en realidad hay algunos comandos disponibles. Vamos a optar por usar un comando del lme4 ( l Inear m ixed correo biblioteca fecto). 3 En nuestro primer uso, instalaremos el paquete y luego, en cada uso, cargaremos la biblioteca:

install.packages (“lme4”)

biblioteca (lme4)

Una vez que hemos cargado la biblioteca, nos ajustamos nuestro modelo lineal multinivel mediante el LMER ( l inear m ixed e FECTOS r egression) comando:

hours.ml <-lmer (hrs\_allev ~ fase1 + senior\_c + ph\_senior + notest\_p +

ph\_notest\_p + female + biocred3 + degr3 + evol\_course + certificado +  
  
 idsci\_trans + confiado + (1 | st\_fip), datos = evolución)

La sintaxis del comando lmer es casi idéntica al código que usamos al ajustar un modelo con OLS usando lm . De hecho, el único atributo que agregamos es el término adicional (1 | st\_fip) en el lado derecho del modelo. Esto agrega una intersección aleatoria por estado. En cualquier ocasión en la que queramos incluir un efecto aleatorio, colocamos entre paréntesis el término para el que se incluye el efecto seguido de una barra vertical y la variable que identifica las unidades de nivel superior. Entonces, en este caso, queríamos una intersección aleatoria (de ahí el uso de 1 ), y queríamos que estos fueran asignados por estado (de ahí el uso de st\_fip ).

Obtenemos nuestros resultados escribiendo:

resumen (horas.ml)

En nuestra salida de resultados, R imprime la correlación entre todos los efectos fijos o parámetros de regresión estimados. Esta parte de la impresión se omite a continuación:

Ajuste de modelo lineal mixto de REML [‘lmerMod’]

Fórmula:

hrs\_allev ~ fase1 + senior\_c + ph\_senior + notest\_p + ph\_notest

\_p +

female + biocred3 + degr3 + evol\_course + certificada + idsci\_  
  
 trans +  
  
seguro + (1 | st\_fip)

Datos: evolución

Criterio REML en la convergencia: 5940

Residuos escalados:

Mín. 1T Mediana 3T Máx.

-2,3478 -0,7142 -0,1754 0,5566 3,8846

Efectos aleatorios:

Grupos Nombre Varianza Desv. Estándar

st\_fip (Intercepción) 3.089 1.758

Residual 67.873 8.239

Número de obs: 841, grupos: st\_fip, 49

Efectos fijos:

Estimar Std. Valor t de error

(Intercepción) 10.5676 1.2138 8.706

fase1 0,7577 0,4431 1,710

senior\_c -0.5291 0.3098 -1.708

ph\_senior -0.5273 0.2699 -1.953

notest\_p 0.1134 0.7490 0.151

ph\_notest\_p -0.5274 0.6598 -0.799

mujer -0,9702 0,6032 -1,608

biocred3 0.5157 0.5044 1.022

grados3 -0,4434 0,3887 -1,141

evol\_course 2.3894 0.6270 3.811

certificado -0,5335 0,7188 -0,742

idsci\_trans 1.7277 1.1161 1.548

seguro 2.6739 0.4468 5.984

La salida primero imprime una variedad de estadísticas de ajuste: AIC, BIC, log-verosimilitud, desviación y desviación de máxima verosimilitud restringida. En segundo lugar, imprime la varianza y la desviación estándar de los efectos aleatorios. En este caso, la varianza para el término st\_fip es la varianza de nuestros efectos aleatorios a nivel de estado. La varianza residual corresponde a la varianza del error de regresión que normalmente calcularíamos para nuestros residuos. Por último, los efectos fijos que se informan son sinónimos de coeficientes de regresión lineal que normalmente nos interesan, aunque ahora nuestras estimaciones han tenido en cuenta la correlación intraclase entre profesores dentro del mismo estado. Cuadro 8.1compara nuestras estimaciones OLS y multinivel una al lado de la otra. Como puede verse, el modelo multinivel ahora divide la varianza inexplicable en dos componentes (nivel estatal e individual), y las estimaciones de coeficientes han cambiado algo. Cuadro 8.1 Dos modelos de horas de clase dedicadas a la enseñanza de la evolución por profesores de biología de secundaria

OLS

Multi nivel

Parámetro

Estimar

Std. error

Pr (> |  z  |)

Estimar

Std. error

Pr (> |  z  |)

Interceptar

10.2313

1.1905

0,0000

10.5675

1.2138

0,0000

Índice de estándares 2007

0,6285

0.3331

0.0596

0,7576

0.4431

0.0873

Antigüedad (centrada)

−0,5813

0.3130

0.0636

−0,5291

0.3098

0.0876

Estándares × antigüedad

−0,5112

0.2717

0.0603

−0,5273

0.2699

0.0508

Cree que no hay prueba

0.4852

0,7222

0.5019

0,1135

0,7490

0.8795

Estándares × cree que no hay prueba

−0,5362

0,6233

0.3899

−0,5273

0,6598

0.4241

La maestra es mujer

−1,3546

0,6016

0.0246

−0,9703

0,6032

0.1077

Créditos obtenidos en biología (0-2)

0.5559

0.5072

0.2734

0.5157

0.5044

0.3067

Grados en ciencias (0-2)

−0.4003

0.3922

0.3077

−0,4434

0.3887

0.2540

Clase de evolución completada

2.5108

0,6300

0,0001

2.3894

0,6270

0,0001

Tiene certificación normal

−0,4446

0,7212

0.5377

−0,5335

0,7188

0.4580

Se identifica como científico

1.8549

1.1255

0.0997

1.7277

1.1161

0.1216

Experiencia autoevaluada (−1 a +2)

2.6262

0.4501

0,0000

2.6738

0,4468

0,0000

Varianza a nivel de estado

3.0892

Varianza a nivel individual

69.5046

67,8732

Notas : N  = 841. Datos de Berkman y Plutzer (2010)

8.1.2 Regresión logística multinivel Si bien es algo más complejo, la lógica del modelado multinivel también se puede aplicar al estudiar variables dependientes limitadas. Hay dos enfoques amplios para extender los GLM a un marco multinivel: modelos marginales, que tienen una interpretación de población promediada, y modelos lineales mixtos generalizados (GLMM), que tienen una interpretación a nivel individual (Laird y Fitzmaurice 2013, págs. 149-156). Si bien se anima a los lectores a leer más sobre los tipos de modelos disponibles, su estimación y su interpretación, por ahora nos centramos en el proceso de estimación de un GLMM.

En este ejemplo, volvemos a nuestro ejemplo de Sect. 7.1.1 del último capítulo, sobre si un encuestado informó haber votado por el partido en el poder en función de la distancia ideológica del partido. Como Singh ( 2014a) observa, los votantes que hagan su elección en el mismo país-año se enfrentarán a muchas características de la elección que son exclusivas de esa elección. Por lo tanto, es probable que exista una correlación intraclase entre los votantes dentro de la misma elección. Además, el efecto de la ideología en sí puede ser más fuerte en algunas elecciones que en otras: los métodos multinivel, incluidos los GLMM, nos permiten evaluar si existe variación en el efecto de un predictor entre grupos, que es una característica que usaremos.

Volviendo a los detalles del código, si la biblioteca lme4 no está cargada, la necesitamos nuevamente. Además, si los datos de no se cargan, entonces necesitamos cargar la biblioteca externa y el conjunto de datos en sí. Todo esto se logra de la siguiente manera: 4

biblioteca (lme4)

biblioteca (extranjera)

votando <-read.dta (“SinghJTP.dta”)

Basándonos en el modelo de la tabla 7.1 , primero simplemente agregamos una intersección aleatoria a nuestro modelo. La sintaxis para estimar el modelo e imprimir los resultados es:

inc.linear.ml <-glmer (votadainc ~ distanciainc + (1 | cntryyear),

familia = binomio (enlace = "logit"), datos = votación)

resumen (incluido lineal.ml)

Tenga en cuenta que ahora usamos la glmer comando ( g eneralized l inear m ixed e FECTOS r egression). Al usar la opción de familia , podemos usar cualquiera de las funciones de enlace comunes disponibles para el comando glm . Un vistazo al resultado muestra que, además de los efectos fijos tradicionales que reflejan los coeficientes de regresión logística, también se nos presenta la varianza de la intersección aleatoria para el país y el año de la elección:

Ajuste de modelo lineal mixto generalizado por Laplace

aproximación

Fórmula: voteinc ~ distanciainc + (1 | cntryyear)

Datos: votación

Desviación de logLik de AIC BIC

41998.96 42024.62 -20996.48 41992.96

Efectos aleatorios:

Grupos Nombre Varianza Desv. Estándar

cntryyear (intersección) 0.20663 0.45457

Número de obs: 38211, grupos: cntryyear, 30

Efectos fijos:

Estimar Std. Error z valor Pr (> | z |)

(Intercepción) 0.161788717 0.085578393 1.89053 0.058687.

distanciainc -0.501250136 0.008875997 -56.47254 <2e-16 \*\*\*

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05. 0,1 1

Correlación de efectos fijos:

(Intr.)

distanciainc -0.185

Para replicar un modelo más acorde con Singh (2014a), ahora ajustamos un modelo que incluye una intersección aleatoria y un coeficiente aleatorio de distancia ideológica, ambos condicionados por el país y el año de la elección. La sintaxis para estimar este modelo e imprimir el resultado es:

inc.linear.ml.2 <-glmer (votadainc ~ distanciainc +

(distanciainc | cntryyear), family = binomial (link = "logit"),  
  
 datos = votación)

resumen (incluido lineal.ml.2)

Observe que ahora hemos condicionado la variable distanciainc por cntryyear . Esto agrega un coeficiente aleatorio para la distancia ideológica. Además, de forma predeterminada, agregar este efecto aleatorio también agrega una intersección aleatoria. Nuestro resultado en este caso es:

Ajuste de modelo lineal mixto generalizado por Laplace

aproximación

Fórmula: voteinc ~ distanciainc + (distanciainc |

cntryyear)

Datos: votación

Desviación de logLik de AIC BIC

41074 41117-20532 41064

Efectos aleatorios:

Grupos Nombre Varianza Desv. Estándar Corr

cntryyear (intersección) 0,616658 0,78528

distanciainc 0.098081 0.31318 -0.808

Número de obs: 38211, grupos: cntryyear, 30

Efectos fijos:

Estimar Std. Error z valor Pr (> | z |)

(Intercepción) 0.26223 0.14531 1.805 0.0711.

distanciainc -0.53061 0.05816 -9.124 <2e-16 \*\*\*

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05. 0,1 1

Correlación de efectos fijos:

(Intr.)

distanciainc -0.808

Bajo efectos aleatorios, primero vemos la varianza para la intersección aleatoria con referencia a la elección y luego la varianza para el coeficiente con referencia a la elección para la distancia ideológica. Los efectos fijos de los coeficientes de regresión logística también se presentan de la forma habitual. El AIC indica que esta versión del modelo se ajusta mejor que el modelo con solo una intersección aleatoria o el modelo de la Tabla 7.1 que no incluyó efectos aleatorios, ya que el puntaje de 41,074 es menor que el AIC de cualquiera de esos modelos. En resumen, esta discusión debería ofrecer una idea de los tipos de modelos jerárquicos que R puede estimar usando lme4 (Bates et al. 2014).

8.2 Métodos bayesianos con MCMCpack El paquete MCMCpack permite a los usuarios realizar inferencias bayesianas en una variedad de modelos de regresión y modelos de medición comunes. El paquete incluso tiene un comando, MCMCmetrop1R , que construirá una muestra de MCMC a partir de una distribución definida por el usuario utilizando un algoritmo de Metropolis. Se anima a los lectores que deseen aprender más sobre los métodos bayesianos a consultar recursos como: Carlin y Louis (2009), Gelman et al. (2004), Branquias (2008) y Robert (2001).

Como una simple ilustración de cómo funciona el paquete, nos enfocamos en esta sección en algunos de los modelos de regresión comunes que están programados en el paquete. Esto es poderoso para el usuario de R, ya que los investigadores que prefieren informar modelos bayesianos pueden hacerlo fácilmente si la especificación de su modelo se ajusta a una estructura común. Al ilustrar estas técnicas, revisaremos una vez más el modelo lineal de horas de evolución de Berkman y Plutzer (2010) y el modelo de regresión logística del apoyo del partido en el poder de Singh (2014a).

8.2.1 Regresión lineal bayesiana Para estimar nuestro modelo de regresión lineal bayesiano, debemos volver a cargar los datos de la Encuesta Nacional de Maestros de Biología de Escuelas Secundarias, si aún no están cargados:

rm (lista = ls ())

biblioteca (extranjera)

evolución <-read.dta (“BPchap7.dta”)

evolución $ mujer [evolución $ mujer == 9] <- NA

evolución <-subconjunto (evolución,! is.na (mujer))

Con los datos cargados, debemos instalar MCMCpack si este es el primer uso del paquete en la computadora. Una vez instalado el programa, debemos cargar la biblioteca:

install.packages (“MCMCpack”)

biblioteca (MCMCpack)

Ahora podemos usar MCMC para ajustar nuestro modelo de iones de regresión lineal bayesiana con el comando MCMCregress :

mcmc.horas <-MCMCregress (hrs\_allev ~ fase1 + senior\_c + ph\_senior +

notest\_p + ph\_notest\_p + female + biocred3 + degr3 +  
  
 evol\_course + certificado + idsci\_trans + seguro, datos = evolución)

Esté preparado para que la estimación con MCMC generalmente toma más tiempo computacionalmente, aunque los modelos simples como este generalmente terminan con bastante rapidez. Además, debido a que MCMC es una técnica basada en simulación, es normal que los resúmenes de los resultados difieran ligeramente entre las repeticiones. Con este fin, si encuentra diferencias entre sus resultados y los impresos aquí después de usar el mismo código, no debe preocuparse a menos que los resultados sean marcadamente diferentes.

Si bien el código anterior se basa en los valores predeterminados del comando MCMCregress , algunas de las opciones de este comando son esenciales para resaltar: Una característica central de los métodos bayesianos es que el usuario debe especificar las prioridades para todos los parámetros que se estiman. Los valores predeterminados para MCMCregress son valores previos conjugados vagos para los coeficientes y la varianza de las perturbaciones. Sin embargo, el usuario tiene la opción de especificar sus propios antecedentes sobre estas cantidades. 5 Se anima a los usuarios a que revisen estas opciones y otros recursos sobre cómo establecer antecedentes (Carlin y Louis 2009; Gelman y col. 2004; Branquia 2008; Robert 2001). Los usuarios también tienen la opción de cambiar el número de iteraciones en la muestra de MCMC con la opción mcmc y el período de quemado (es decir, el número de iteraciones iniciales que se descartan) con la opción de quemado . Los usuarios siempre deben evaluar la convergencia del modelo después de estimar un modelo con MCMC (que se discutirá en breve) y considerar si el quemado o el número de iteraciones deben cambiarse si hay evidencia de no convergencia.

Después de estimar el modelo, escribiendo resumen (mcmc.hours) ofrecerá un resumen rápido de la muestra posterior :

Iteraciones = 1001: 11000

Intervalo de dilución = 1

Número de cadenas = 1

Tamaño de muestra por cadena = 10000

1. Media empírica y desviación estándar de cada

* variable, más el error estándar de la media:
* Media SD Naive SE Serie temporal SE

(Intercepción) 10,2353 1,1922 0,011922 0,011922

fase1 0,6346 0,3382 0,003382 0,003382

senior\_c -0.5894 0.3203 0.003203 0.003266

ph\_senior -0,5121 0,2713 0,002713 0,002713

notest\_p 0,4828 0,7214 0,007214 0,007214

ph\_notest\_p -0.5483 0.6182 0.006182 0.006182

mujer -1,3613 0,5997 0,005997 0,006354

biocred3 0,5612 0,5100 0,005100 0,005100

grados3 -0,4071 0,3973 0,003973 0,003973

evol\_course 2,5014 0,6299 0,006299 0,005870

certificado -0,4525 0,7194 0,007194 0,007194

idsci\_trans 1.8658 1.1230 0.011230 0.010938

seguro 2.6302 0.4523 0.004523 0.004590

sigma2 70,6874 3,5029 0,035029 0,035619

1. Cuantiles para cada variable:

* 2,5% 25% 50% 75% 97,5%

(Intercepción) 7.92359 9.438567 10.2273 11.03072 12.59214

fase1 -0.02787 0.405026 0.6384 0.86569 1.30085

senior\_c -1.22527 -0.808038 -0.5885 -0.37351 0.04247

ph\_senior -1.04393 -0.694228 -0.5105 -0.32981 0.03152

notest\_p -0.92717 -0.006441 0.4863 0.97734 1.88868

ph\_notest\_p -1.75051 -0.972112 -0.5462 -0.13138 0.63228

femenino -2.52310 -1.771210 -1.3595 -0.96109 -0.18044

biocred3 -0,42823 0,212168 0,5558 0,90768 1,55887

grados3 -1.19563 -0.671725 -0.4048 -0.14536 0.38277

evol\_course 1.26171 2.073478 2.5064 2.92601 3.73503

certificado -1.84830 -0.942671 -0.4477 0.03113 0.95064

idsci\_trans -0.33203 1.107771 1.8667 2.63507 4.09024

confiado 1.73568 2.324713 2.6338 2.94032 3.48944

sigma2 64.12749 68.277726 70.5889 72.95921 77.84095

Dado que MCMC produce una muestra de valores de parámetros simulados, toda la información reportada se basa en estadísticas descriptivas simples de la salida simulada (que son 10,000 conjuntos de parámetros, en este caso). La parte 1 del resumen anterior muestra la media de la muestra para cada parámetro, la desviación estándar de la muestra de cada parámetro y dos versiones del error estándar de la media. La parte 2 del resumen muestra los percentiles de la muestra de cada parámetro. La Tabla 8.2 muestra un formato común para presentar los resultados de un modelo como este: informar la media y la desviación estándar de la distribución posterior marginal de cada parámetro y un intervalo de credibilidad del 95% basado en los percentiles. 6 Cuadro 8.2 Modelo lineal de horas de clase dedicadas a la evolución de la enseñanza por profesores de biología de secundaria (estimaciones de MCMC)

Vaticinador

Significar

Std. Dev.

[95% Cred. En t.]

Interceptar

10.2353

1.1922

[7,9236:

12.5921]

Índice de estándares 2007

0,6346

0.3382

[−0,0279:

1.3008]

Antigüedad (centrada)

−0,5894

0.3203

[−1,2253:

0.0425]

Estándares × antigüedad

−0,5121

0.2713

[−1.0439:

0.0315]

Cree que no hay prueba

0.4828

0,7214

[−0,9272:

1.8887]

Estándares × cree que no hay prueba

−0,5483

0,6182

[−1,7505:

0.6323]

La maestra es mujer

−1,3613

0.5997

[−2,5231:

−0,1804]

Créditos obtenidos en biología (0-2)

0.5612

0.5100

[−0,4282:

1.5589]

Grados en ciencias (0-2)

−0,4071

0.3973

[−1,1956:

0.3828]

Clase de evolución completada

2.5014

0,6299

[1.2617:

3.7350]

Tiene certificación normal

−0,4525

0,7194

[−1,8483:

0.9506]

Se identifica como científico

1.8658

1.1230

[−0,3320:

4.0902]

Experiencia autoevaluada (- 1 a + 2)

2.6302

0.4523

[1.7357:

3.4894]

Varianza de error de regresión

70.6874

3.5029

[64,1275:

77.8410]

Notas : N  = 841. Datos de Berkman y Plutzer (2010)

Cuando un usuario carga MCMCpack en R , también se cargará la biblioteca de coda . 7 coda es particularmente útil porque permite al usuario evaluar la convergencia de los modelos estimados con MCMC y reportar cantidades adicionales de interés. Como se mencionó anteriormente, cada vez que un investigador estima un modelo utilizando MCMC, debe evaluar si existe alguna evidencia de no convergencia. En el caso de que las cadenas de Markov no hayan convergido, el modelo debe ser muestreado para más iteraciones. MCMCregress estima el modelo utilizando una sola cadena. Por lo tanto, para nuestro modelo del número de horas de evolución que se enseñan en las aulas de secundaria, podemos evaluar la convergencia utilizando Geweke convergencia ‘sdiag nostic, que simplemente pregunta si los medios de la primera y última parte de la cadena son los mismos. Para calcular este diagnóstico, escribimos:

geweke.diag (mcmc.horas, frac1 = 0.1, frac2 = 0.5)

Aquí hemos especificado que queremos comparar la media del primer 10% de la cadena ( frac1 = 0.1 ) con la media del último 50% de la cadena ( frac2 = 0.5 ). La salida resultante presenta una relación z para esta prueba de diferencia de medias para cada parámetro:

Fracción en la 1ra ventana = 0.1

Fracción en la segunda ventana = 0.5 (Intercepción) fase1 senior\_c ph\_senior notest\_p

-1,34891 -1,29015 -1,10934 -0,16417 0,95397

ph\_notest\_p hembra biocred3 degr3 evol\_course

1,13720 -0,57006 0,52718 1,25779 0,62082

certificado idsci\_trans seguro sigma2

1,51121 -0,87436 -0,54549 -0,06741

En este caso, ninguna de las estadísticas de prueba supera ningún umbral de significancia común para una estadística de prueba distribuida normalmente, por lo que no encontramos evidencia de no convergencia. Con base en esto, podemos estar contentos con nuestra muestra original de MCMC de 10,000.

Una cosa más que podemos desear hacer con nuestra salida de MCMC es trazar la función de densidad estimada general de nuestras distribuciones posteriores marginales. Podemos graficar estos uno a la vez utilizando la función densplot , aunque el analista deberá hacer referencia al parámetro de interés según su orden numérico de aparición en la tabla de resumen. Por ejemplo, si quisiéramos graficar el coeficiente para el indicador de si un docente completó una clase de evolución ( evol\_course ), esa es la estimación del décimo parámetro reportado en la tabla. De manera similar, si quisiéramos informar la gráfica de densidad para el coeficiente de la experiencia autoevaluada del maestro ( confianza), que es el decimotercer parámetro informado en la tabla resumen. Por lo tanto, podríamos trazar cada uno de estos escribiendo:

densplot (mcmc.horas [, 10])

densplot (mcmc.horas [, 13])

Las gráficas de densidad resultantes se presentan en la Fig.  8.1 . Como muestran las figuras, ambas distribuciones posteriores marginales tienen una distribución normal aproximada y la moda se encuentra cerca de la media y la mediana informadas en nuestro resultado resumido. Abrir imagen en nueva ventanaFigura 8.1 Figura 8.1 Gráficos de densidad de distribución marginal posterior de coeficientes para determinar si el maestro completó una clase de evolución y la pericia autoevaluada del maestro. Basado en una muestra de MCMC de 10,000 iteraciones (quemado de 1000 iteraciones). ( a ) Clase de evolución completa; ( b ) Experiencia autoevaluada

8.2.2 Regresión logística bayesiana Como una ilustración adicional de que MCMCpack estima una variedad de modelos, ilustramos la regresión logística bayesiana, utilizando Singh (2014a) datos sobre la votación de los partidos en el poder por última vez. Si no tiene estos datos o la biblioteca cargada, asegúrese de hacerlo:

biblioteca (MCMCpack)

biblioteca (extranjera)

votando <-read.dta (“SinghJTP.dta”)

Para estimar el modelo logit bayesiano usando MCMC , escribimos :

inc.linear.mcmc <-MCMClogit (votadainc ~ distanciainc, datos = votando)

Al igual que con el comando MCMCregress , hemos optado por utilizar los valores predeterminados en este caso, pero se recomienda a los usuarios que consideren establecer sus propios valores previos para satisfacer sus necesidades. De hecho, este es un caso en el que necesitaremos aumentar el número de iteraciones en nuestro modelo. Podemos verificar la convergencia de nuestro modelo usando el diagnóstico de Geweke:

geweke.diag (incluido linear.mcmc, frac1 = 0.1, frac2 = 0.5)

Nuestro resultado en este caso muestra una diferencia significativa entre las medias al principio y al final de la cadena para cada parámetro:

Fracción en la 1ra ventana = 0.1

Fracción en la segunda ventana = 0.5

(Intercepción) distanciainc

2.680 -1.717

El valor absoluto de ambas razones z excede 1.645, por lo que podemos decir que la media es significativamente diferente para cada parámetro al nivel de confianza del 90%, lo cual es evidencia de no convergencia.

Como respuesta, podemos duplicar nuestro período de quemado y el número de iteraciones a 2,000 y 20,000, respectivamente. El código es:

inc.linear.mcmc.v2 <-MCMClogit (votadoinc ~ distanciainc,

datos = votación, quemado = 2000, mcmc = 20000)

Ahora podemos verificar la convergencia de esta nueva muestra escribiendo:

geweke.diag (incluido linear.mcmc.v2, frac1 = 0.1, frac2 = 0.5)

Nuestra salida ahora muestra relaciones z no significativas para cada parámetro, lo que indica que ya no hay evidencia de no convergencia:

Fracción en la 1ra ventana = 0.1

Fracción en la segunda ventana = 0.5

(Intercepción) distanciainc

-1,0975 0,2128

Continuando con esta muestra de 20.000, entonces, si escribimos

resumen (inc.linear.mcmc.v2) en la consola, el resultado es:

Iteraciones = 2001: 22000

Intervalo de dilución = 1

Número de cadenas = 1

Tamaño de muestra por cadena = 20000

1. Media empírica y desviación estándar de cada

* variable, más el error estándar de la media:
* Media SD Naive SE Serie temporal SE

(Intercepción) 0.1940 0.01846 1.305e-04 0.0003857

distanciainc -0.4946 0.00829 5.862e-05 0.0001715

1. Cuantiles para cada variable:

* 2,5% 25% 50% 75% 97,5%

(Intercepción) 0,1573 0,1817 0,1944 0,2063 0,2298

distanciainc -0.5105 -0.5003 -0.4946 -0.4890 -0.4783

Estos resultados son similares a los que informamos en el último capítulo en la tabla 7.1 , aunque ahora tenemos la oportunidad de interpretar los hallazgos como bayesianos. Al igual que con la regresión lineal bayesiana, si quisiéramos informar las gráficas de densidad de cada parámetro, podríamos aplicar el comando densplot como antes. En general, esta breve ilustración debería mostrar a los investigadores lo fácil que es usar métodos bayesianos en R con MCMCpack . Se recomienda a los lectores que deseen utilizar métodos bayesianos más avanzados que consulten el manual de referencia de MCMCpack y Martin et al. ( 2011).

8.3 Inferencia causal con cem Una innovación destacada en la metodología política ha sido el desarrollo de varios métodos nuevos de emparejamiento. En resumen, el emparejamiento es una técnica diseñada para seleccionar un subconjunto de datos de campo para realizar una comparación justa de las personas que reciben un tratamiento para controlar a las personas que no recibieron el tratamiento. Con el emparejamiento, algunas observaciones se descartan para que las observaciones de control y tratamiento restantes sean similares en todas las covariables que se sabe que dan forma al resultado de interés. En ausencia de datos experimentales, los métodos de emparejamiento sirven para permitir al investigador aislar cómo la variable de tratamiento afecta las respuestas (ver Rubin 2006 para un tratamiento completo de la inferencia causal con emparejamiento).

Los científicos políticos han desarrollado varios métodos nuevos de emparejamiento (Imai y van Dyk 2004; Sekhon y Grieve 2012). Como ilustración de uno de ellos, y de cómo se implementa la técnica novedosa en R , recurrimos al método desarrollado por Iacus et al. (2009, 2011, 2012), C oarsened E xact M atching (CEM). En resumen, CEM procede recodificando temporalmente cada covariable en una variable ordenada que agrupa valores similares de la covariable. Luego, los datos se clasifican en estratos en función de sus perfiles de las variables aproximadas, y cualquier observación en un estrato que no contiene al menos un tratamiento y una unidad de control se desecha. La muestra resultante debe mostrar mucho más equilibrio en las variables de control entre las observaciones tratadas y de control. En R , esta técnica se implementa con el comando cem dentro del paquete cem .

8.3.1 Desequilibrio de covariables, implementación de CEM y ATT Como datos de nuestro ejemplo, volvemos a LaLonde (1986) estudio de la Demostración Nacional de Trabajo Apoyado que examinamos anteriormente en los Capítulos. 4 y 5. Nuestra variable de tratamiento ( tratado ) indica si la persona recibió el tratamiento de la Demostración Nacional de Trabajo Apoyado, es decir, la persona fue colocada en un trabajo del sector privado durante un año con fondos públicos que cubren los costos laborales. Nos gustaría conocer el efecto causal de este tratamiento sobre los ingresos del individuo en 1978, una vez finalizado el tratamiento ( re78 ). En la Secta. 5.1.1 Hicimos una prueba ingenua de esta hipótesis simplemente usando una prueba de diferencia de medias entre los individuos tratados y el grupo de control sin controlar ninguna de las otras variables que probablemente afecten el ingreso anual. En nuestra prueba ingenua, encontramos que los ingresos eran más altos para el grupo tratado, pero ahora podemos preguntar cuál es el efecto estimado cuando contabilizamos otras covariables importantes.

Como se señaló en capítulos anteriores, los datos de LaLonde ya están incluidos en el paquete cem , por lo que podemos cargar estos datos fácilmente si ya hemos cargado la biblioteca cem . Las siguientes líneas de código limpian, instalan cem (en caso de que aún no lo haya hecho), abren cem y cargan los datos de LaLonde (llamados LL ): 8

rm (lista = ls ())

install.packages (“cem”)

biblioteca (cem)

datos (LL)

Una tarea importante cuando se utilizan métodos de emparejamiento es evaluar el grado en que los datos están equilibrados o el grado en que los casos tratados tienen una distribución similar de valores de covariables en relación con el grupo de control. Podemos evaluar el grado en que nuestros grupos de tratamiento y control tienen distribuciones diferentes con el comando de desequilibrio . En el siguiente código, primero aumentamos la penalización por la notación científica (una opción si prefiere la notación decimal). Luego, creamos un vector que nombra las variables para las que no queremos evaluar el equilibrio: la variable de tratamiento ( tratada ) y el resultado de interés ( re78). Todas las demás variables del conjunto de datos son covariables que creemos que pueden dar forma a los ingresos en 1978, por lo que nos gustaría tener un equilibrio sobre ellas. En la última línea, en realidad llamamos al comando de desequilibrio .

opciones (scipen = 8)

todrop <- c (“tratado,” “re78”)

desequilibrio (grupo = LL $ tratado, datos = LL, drop = todrop)

Dentro del comando de desequilibrio , el argumento de grupo es nuestra variable de tratamiento que define los dos grupos para los que queremos comparar las distribuciones de covariables. El argumento de datos nombra el conjunto de datos que estamos usando y la opción de descartar nos permite omitir ciertas variables del conjunto de datos al evaluar el equilibrio de covariables. Nuestro resultado de este comando es el siguiente:

Medida de desequilibrio multivariante: L1 = 0,735

Porcentaje de apoyo común local: LCS = 12,4%

Medidas de desequilibrio univariante:

tipo de estadística L1 min 25%

edad 0.179203803 (diff) 4.705882e-03 0 1

educación 0.192236086 (diff) 9.811844e-02 1 0

negro 0.001346801 (diff) 1.346801e-03 0 0

casado 0.010703110 (diff) 1.070311e-02 0 0

nodegree -0.083477916 (diff) 8.347792e-02 0-1

re74 -101.486184085 (diff) 5.551115e-17 0 0

re75 39.415450601 (diff) 5.551115e-17 0 0

hispano -0.018665082 (diff) 1.866508e-02 0 0

u74 -0.020099030 (diferencia) 2.009903e-02 0 0

u75 -0.045086156 (diferencia) 4.508616e-02 0 0

50% 75% máximo

edad 0.00000 -1.0000 -6.0000

educación 1,00000 1,0000 2,0000

negro 0,00000 0,0000 0,0000

casado 0,00000 0,0000 0,0000

grado de nodo 0,00000 0,0000 0,0000

re74 69.73096 584.9160 -2139.0195

re75 294.18457 660.6865 490.3945

hispano 0,00000 0,0000 0,0000

u74 0,00000 0,0000 0,0000

u75 0,00000 0,0000 0,0000

La primera línea de nuestros informes de salida L1, que es una medida del desequilibrio multivariado creado por Iacus et al. (2011). En ese artículo se encuentra disponible una explicación más completa, pero en general esta estadística varía de 0 a 1, y los valores más bajos indican un mejor equilibrio. Cuándo L1= 0 las dos distribuciones se superponen perfectamente, y cuando L1= 1las dos distribuciones no se superponen en absoluto. Volviendo a la tabla, cada fila muestra varias estadísticas de saldo para una covariable individual. Para todas estas estadísticas, los valores más cercanos a cero son mejores. La columna etiquetada como estadístico muestra la diferencia de medias entre las variables, y la columna etiquetada como L1 calcula Lj1, que es la misma medida que L1pero solo calculado para la covariable individual. Las columnas restantes muestran diferencias de cuantiles entre los dos grupos (por ejemplo, la diferencia en los mínimos respectivos de los dos grupos, la diferencia entre los percentiles 25 respectivos de los grupos, etc.).

A continuación, vamos a utilizar realmente el CEM de comandos para realizar C oarsened E xact M atching en nuestros datos. Dentro del comando cem , enumeramos nuestra variable de tratamiento con el argumento de tratamiento , nuestro conjunto de datos con el argumento de datos y cualquier variable que no queramos que coincida con el argumento de caída . La caída argumento debe incluir siempre nuestra variable de resultado, si está en el mismo conjunto de datos, así como los índices de datos o variables irrelevantes. Podríamos usar un vector para enumerar todas las variables que queremos que se ignoren, como hicimos antes con el comando de desequilibrio , pero en este caso, solo el resultadore78 debe omitirse . Escribimos:

cem.match.1 <- cem (tratamiento = “tratado,” datos = LL, gota = “re78”)

cem.match.1

Nuestro resultado inmediato de esto es simplemente el siguiente:

G0 G1

Todos 425297

Emparejado 222163

Inigualable 203134

Esto nos dice que nuestros datos originales tenían 425 observaciones de control y 297 observaciones tratadas. El CEM incluyó 222 del control y 163 de las observaciones tratadas en la muestra emparejada, y el resto se eliminó. Para ser claros: todas las observaciones todavía están contenidas en el conjunto de datos LL original , pero ahora el objeto cem.match.1 detalla qué observaciones coinciden o no.

Dado que CEM procede agrupando valores similares de covariables en estratos, una característica importante de esto es cómo establecemos los intervalos ordenados de cada predictor en el engrosamiento. El comando cem tiene valores predeterminados razonables si el usuario no establece estos intervalos, pero es importante registrar cuáles son los intervalos. Para ver cuáles fueron nuestros intervalos para los valores de nuestros predictores, podríamos escribir: cem.match.1 $ breaks . Esto nos daría el siguiente resultado:

$ edad

[1] 17,0 20,8 24,6 28,4 32,2 36,0 39,8 43,6 47,4 51,2 55,0

$ educación

[1] 3,0 4,3 5,6 6,9 8,2 9,5 10,8 12,1 13,4 14,7 16,0

$ negro

[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

$ casado

[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

$ nodegree

[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

$ re74

[1] 0.000 3957.068 7914.136 11871.204 15828.272 19785.340

[7] 23742.408 27699.476 31656.544 35613.612 39570.680

$ re75

[1] 0.000 3743.166 7486.332 11229.498 14972.664 18715.830

[7] 22458.996 26202.162 29945.328 33688.494 37431.660

$ hispano

[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

$ u74

[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

$ u75

[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

Para ilustrar lo que esto significa, considere la edad . La categoría más baja de edad engrosada agrupa a todas las personas de 17 a 20,8 años, la segunda categoría agrupa a todas las personas de 20,8 a 24,6 años, y así sucesivamente. Variables como negro , casado , nodal , hispano , u74 y u75 son en realidad binarias, por lo que la mayoría de las categorías que se crean son innecesarias, aunque los contenedores vacíos no dañarán nuestro análisis. Por supuesto, los usuarios no están obligados a utilizar los valores predeterminados de software, y Iacus et al. instar a los investigadores a utilizar un conocimiento sustancial de la medición de cada variable para establecer los rangos de los contenedores de engrosamiento (2012, pag. 9). La sección  8.3.2 ofrece detalles sobre cómo hacer esto.

Ahora podemos evaluar el desequilibrio en la nueva muestra emparejada escribiendo:

desequilibrio (LL $ tratado [cem.match.1 $ igualado],

LL [cem.match.1 $ coincidente,], drop = todrop)

Nuestro resultado de esto es el siguiente: Medida de desequilibrio multivariante: L1 = 0.592

Porcentaje de apoyo común local: LCS = 25,2%

Medidas de desequilibrio univariante:

tipo de estadística L1 min 25% 50%

edad -0.42486044 (diff) 0.00000000 0-1 -2.0000

educación -0.10855027 (diff) 0.10902006 0 0 -1.0000

negro -0.01771403 (diff) 0.01771403 0 0 0.0000

casado -0.01630465 (diff) 0.01630465 0 0 0.0000

grado de nodo 0.09022827 (diff) 0.09022827 0 0 0.0000

re74 -119.33548135 (diff) 0.00000000 0 0 0.0000

re75 -50.01527694 (diff) 0.00000000 0 0 -49.3559

hispano 0.01561377 (diff) 0.01561377 0 0 0.0000

u74 0,01619411 (diferencia) 0,01619411 0 0 0,0000

u75 0,02310286 (dif.) 0,02310286 0 0 0,0000

75% máximo

edad 0,00 1.000

educación 0,00 0,000

negro 0,00 0,000

casado 0,00 0,000

grado de nodo 0,00 0,000

re74 -492.95 416.416

re75 -136.45 -852.252

hispano 0,00 0,000

u74 0,00 0,000

u75 0,00 0,000

Compare esto con los datos originales. Ahora tenemos L1= 0,592, que es menor que nuestra puntuación de 0,735 para los datos brutos, lo que indica que el equilibrio multivariado es mejor en la muestra emparejada. En cuanto a las covariables individuales, puede ver una especie de bolsa mixta, pero en general el equilibrio se ve mejor. Por ejemplo, con la edad, la diferencia de medias es en realidad un poco mayor en valor absoluto con la muestra emparejada (0,42) que con los datos brutos (0,18). Sin embargo, La g e1ahora es minúsculo en la muestra emparejada y menor que el valor 0.0047 para los datos sin procesar. Esto probablemente se deba al hecho de que los datos brutos tienen una mayor discrepancia en el extremo superior que la muestra emparejada. El usuario ahora debe decidir si los casos tratados y de control están lo suficientemente equilibrados o si probar otros engrosamientos para mejorar el equilibrio.

Si el usuario está satisfecho con el nivel de equilibrio, él o ella puede proceder a estimar el A verage T ratamiento efecto sobre la T reated (ATT) con el comando att . Esta cantidad representa el efecto causal sobre el tipo de individuo que recibió el tratamiento. En este comando especificamos cuál es nuestra muestra emparejada usando el argumento obj , la variable de resultado ( re78 ) y el tratamiento ( tratado ) usando el argumento de fórmula , y nuestros datos usando el argumento de datos . Esto nos da:

est.att.1 <- att (obj = cem.match.1, fórmula = re78 ~ tratado, datos = LL)

est.at.1

Nuestro resultado de esto es:

G0 G1

Todos 425297

Emparejado 222163

Inigualable 203134

Modelo de regresión lineal sobre datos emparejados CEM:

Estimación puntual SATT: 550,962564 (valor p = 0,368242)

95% conf. intervalo: [-647.777701, 1749.702830]

El resultado resume las características de nuestra muestra emparejada y luego informa nuestra estimación del efecto del tratamiento promedio de la muestra en los tratados (SATT): Estimamos en nuestra muestra que los individuos que recibieron el tratamiento ganaron $ 551 más en promedio que aquellos que no recibieron el tratamiento. . Sin embargo, este efecto no es estadísticamente discernible de cero ( p  = 0. 368). Ésta es una conclusión marcadamente diferente de la que extrajimos en el cap. 5 , cuando observamos una diferencia de $ 886 que fue estadísticamente significativa. Esto ilustra la importancia del control estadístico.

8.3.2 Explorando diferentes soluciones CEM Como punto final, si un investigador no está satisfecho con el nivel de equilibrio o el tamaño de la muestra en la muestra emparejada, entonces una herramienta para encontrar un mejor equilibrio es el comando cemspace . Este comando produce aleatoriamente varios grosores diferentes para las variables de control (250 diferentes grosores por defecto). Luego, el comando traza el nivel de equilibrio contra el número de observaciones tratadas incluidas en la muestra emparejada. El siguiente código llama a este comando:

cem.explore <-cemspace (tratamiento = “tratado,” datos = LL, gota = “re78”)

La sintaxis de cemspace es similar a cem , aunque dos opciones más son importantes: mínima y máxima . Estos establecen cuál es el número mínimo y máximo permitido de intervalos aproximados para las variables. El comando anterior usa los valores predeterminados de 1 y 5, lo que significa que no se pueden incluir más de cinco intervalos para una variable. Por lo tanto, todas las muestras coincidentes de este comando serán más burdas que las que usamos en la sección. 8.3.1 y, por tanto, menos equilibrado. Sin embargo, el usuario podría aumentar el máximo a 12 o incluso un número más alto para crear intervalos más finos y potencialmente mejorar el equilibrio con respecto a nuestro resultado anterior.

Nuestra salida de cemspace se muestra en la figura  8.2 . Debido al elemento aleatorio en la elección de grosores, sus resultados no coincidirán exactamente con esta cifra. La figura 8.2 a muestra la figura interactiva que se abre. El eje horizontal de esta figura muestra el número de unidades de tratamiento emparejadas en orden descendente, mientras que el eje vertical muestra el nivel de desequilibrio. En general, una muestra emparejada en la esquina inferior izquierda del gráfico sería ideal, ya que indicaría el mejor equilibrio (reducción del sesgo) y la muestra más grande (aumento de la eficiencia). Normalmente, sin embargo, tenemos que tomar una decisión sobre esta compensación, generalmente poniendo un poco más de peso en minimizar el desequilibrio. Al hacer clic en diferentes puntos del gráfico, la segunda ventana quecemspace crea, que se muestra en la figura  8.2 b mostrará los intervalos utilizados en ese engrosamiento particular. El usuario puede copiar los vectores de los puntos de corte del intervalo y pegarlos en su propio código. Nota: R no continuará con nuevos comandos hasta que estas dos ventanas estén cerradas. Abrir imagen en nueva ventanaFigura 8.2 Figura 8.2 Gráfica de las estadísticas de equilibrio para 250 muestras emparejadas de groserías aleatorias contra el número de observaciones tratadas incluidas en la muestra emparejada respectiva. ( a ) Ventana de trazado; ( b ) Ventana X11

En la Fig.  8.2 a, se ha elegido uno de los posibles grosores, y está resaltado y en amarillo. Si queremos implementar este engrosamiento, podemos copiar los vectores que se muestran en la segunda ventana ilustrada en la figura  8.2 b. Pegarlos en nuestro propio script R produce el siguiente código:

age.cut <-c (17, 26.5, 36, 45.5, 55)

educación.corte <-c (3, 6.25, 9.5, 12.75, 16)

black.cut <-c (0, 0.5, 1)

casado.cut <-c (0, 0.5, 1)

nodogree.cut <-c (0, 0.5, 1)

re74.cut <-c (0, 19785.34, 39570.68)

re75.cut <-c (0, 9357.92, 18715.83, 28073.75, 37431.66)

corte.hispano <-c (0, 0.5, 1)

u74.cut <-c (0, 0.5, 1)

u75.cut <-c (0, 1)

new.cuts <-list (age = age.cut, education = education.cut,

negro = corte negro, casado = corte casado,  
  
 nodegree = nodegree.cut, re74 = re74.cut, re75 = re75.cut,  
  
 hispano = hispano.cut, u74 = u74.cut, u75 = u75.cut)

Terminamos este código creando una lista de todos estos vectores. Si bien nuestros vectores aquí se han creado utilizando un engrosamiento creado por cemspace , este es el procedimiento que usaría un programador para crear sus propios puntos de corte para los intervalos. Sustituyendo los vectores anteriores con vectores de punto de corte creados por el usuario, un investigador puede usar su propio conocimiento de la medición de las variables para engrosar.

Una vez que hemos definido nuestros propios puntos de corte , ya sea usando cemspace o conocimiento sustantivo, ahora podemos aplicar CEM con el siguiente código:

cem.match.2 <- cem (tratamiento = “tratado,” datos = LL, gota = “re78,”

puntos de corte = new.cuts)

Nuestra adición clave aquí es el uso de la opción de puntos de corte , donde ingresamos nuestra lista de intervalos. Al igual que en la Secta. 8.3.1 , ahora podemos evaluar las cualidades de la muestra emparejada, los niveles de desequilibrio y calcular el ATT si lo deseamos:

cem.match.2

desequilibrio (LL $ tratado [cem.match.2 $ igualado],

LL [cem.match.2 $ coincidente,], drop = todrop)

est.att.2 <- att (obj = cem.match.2, fórmula = re78 ~ tratado, datos = LL)

est.at.2

En este caso, en parte debido a los bins más gruesos que estamos usando, el saldo es peor que el que encontramos en la sección anterior. Por lo tanto, estaríamos mejor en este caso si nos quedamos con nuestro primer resultado. Se anima al lector a intentar encontrar un engrosamiento que produzca un mejor equilibrio que los valores predeterminados del software.

8.4 Análisis de lista legislativa con wnominate Metodólogos en Ciencias Políticas y otras disciplinas han desarrollado una amplia gama de modelos de medición, varios de los cuales están disponibles para la aplicación de usuario en R . Sin duda, una de las mayoría de los modelos de medición prominentes en la disciplina es designar, de corto para nomina l t hree paso e stimation (Poole y Rosenthal 1997; McCarty y col. 1997; Poole y col. 2011). El modelo NOMINATE analiza los datos de la votación nominal de las votaciones legislativas, ubicando a los legisladores y las alternativas políticas que votan en el espacio ideológico. El modelo es particularmente prominente porque Poole, Rosenthal y sus colegas hacen que los puntajes DW-NOMINATE estén disponibles tanto para la Cámara de Representantes como para el Senado de EE. UU. Para cada período del Congreso. Innumerables autores han utilizado estos datos, interpretando típicamente la puntuación de la primera dimensión como una escala de ideología liberal-conservadora.

En resumen, la lógica básica del modelo se basa en el modelo de proximidad espacial de la política, que esencialmente establece que tanto las preferencias ideológicas de los individuos como las alternativas políticas disponibles pueden representarse en un espacio geométrico de una o más dimensiones. Un individuo generalmente votará por la opción de política que más se acerque en el espacio a su propio punto ideológico ideal (Hotelling 1929; Downs 1957; Negro 1958). El modelo NOMINATE se basa en estos supuestos y coloca a los legisladores y las opciones de políticas en el espacio ideológico según la forma en que los votos de los legisladores se dividen en el transcurso de muchas votaciones nominales y cuando los legisladores se comportan de manera impredecible (produciendo errores en el modelo). Por ejemplo, en el Congreso de los Estados Unidos, los miembros liberales suelen votar de manera diferente a los miembros conservadores, y es más probable que los ideólogos extremos estén en una pequeña minoría siempre que haya un amplio consenso sobre un tema. Antes de aplicar el método NOMINATE a sus propios datos, e incluso antes de descargar datos DW-NOMINATE premedidos para incluirlos en un modelo que calcule, asegúrese de leer más sobre el método y sus suposiciones porque comprender a fondo cómo funciona un método es esencial antes usándolo. En particular, el Cap. 2 y el Apéndice A de Poole y Rosenthal (1997) y el Apéndice A de McCarty et al. (1997) ofrecen descripciones detalladas, pero intuitivas, de cómo funciona el método.

En R , el paquete wnominate implementa W-NOMINATE, que es una versión del algoritmo NOMINATE que solo debe aplicarse a una sola legislatura. Los puntajes de W-NOMINATE son válidos internamente, por lo que es justo comparar los puntajes de los legisladores dentro de un solo conjunto de datos. Sin embargo, los puntajes no se pueden comparar externamente con los puntajes cuando W-NOMINATE se aplica a un período diferente de la legislatura oa un cuerpo de actores completamente diferente. Por tanto, es un buen método para intentar hacer comparaciones transversales entre legisladores de un mismo organismo.

Si bien la aplicación más común para W-NOMINATE ha sido el Congreso de los Estados Unidos, el método podría aplicarse a cualquier cuerpo legislativo. Con ese fin, el ejemplo práctico de esta sección se centra en las votaciones nominales emitidas en las Naciones Unidas. Este conjunto de datos de la ONU está disponible en el paquete wnominate y reúne 237 votaciones nominales emitidas en las tres primeras sesiones de la ONU (1946-1949) por 59 países miembros. Las variables están etiquetadas como V1 a V239 . V1 es el nombre de la nación miembro y V2 es una variable categórica codificada como “WP” para un miembro del Pacto de Varsovia, u “Otro” para todas las demás naciones. Las variables restantes identifican secuencialmente las votaciones nominales.

Para comenzar, limpiamos, instalamos el paquete wnominate la primera vez que lo usamos, cargamos la biblioteca y cargamos los datos UN : 9

rm (lista = ls ())

install.packages (“wnominate”)

biblioteca (wnominate)

datos (ONU)

Una vez que se cargan los datos, se pueden ver con los comandos estándar como fix , pero para una vista rápida de cómo se ven los datos, simplemente podríamos escribir: head (UN [, 1: 15]) . Esto mostrará la estructura de los datos a través de las primeras 13 votaciones nominales.

Antes de que podamos aplicar W-NOMINATE, tenemos que reformatear los datos a un objeto de clase rollcall . Para hacer esto, primero tenemos que redefinir nuestro conjunto de datos de la ONU como una matriz y dividir los nombres de los países, si el país estaba en el Pacto de Varsovia y el conjunto de llamadas de lista en tres partes separadas:

UN <-as.matrix (UN)

UN.2 <-UN [, - c (1,2)]

UNnames <-UN [, 1]

legData <-matrix (UN [, 2], length (UN [, 2]), 1)

La primera línea convirtió el marco de datos de la ONU en una matriz. (Para obtener más información sobre los comandos de la matriz en R , consulte el capítulo  10 ) La segunda línea creó una nueva matriz, que hemos denominado UN.2 , que ha eliminado las dos primeras columnas (nombre del país y miembro del Pacto de Varsovia) para dejar solo el pasa lista. La tercera línea exporta los nombres de las naciones al vector UNnames . (En muchos otros entornos, este sería el nombre del legislador o una variable de identificación). Por último, nuestra variable de si una nación estaba en el Pacto de Varsovia se ha guardado como una matriz de una columna denominada legData . (En muchos otros entornos, este sería el partido político de un legislador). Una vez que tengamos estos componentes juntos, podemos usar el comando rollcall para definir un objeto rollcall -class que llamamos rc de la siguiente manera:

rc <-rollcall (data = UN.2, yea = c (1,2,3), nay = c (4,5,6),

missing = c (7,8,9), notInLegis = 0, legis.names = UNnames,  
  
 legis.data = legData, desc = "UN Votes", source = "voteview.com")

Especificamos nuestra matriz de votaciones nominales con el argumento de datos . Basándonos en cómo se codifican los datos en la matriz de votación nominal, usamos los argumentos sí , no y faltantes para traducir los códigos numéricos a su significado sustantivo. Además, notInLegis nos permite especificar un código que significa específicamente que el legislador no era miembro en el momento de la votación (por ejemplo, un legislador falleció o renunció). No tenemos tal caso en estos datos, pero el valor predeterminado es notInLegis = 9 , y 9 significa algo más para nosotros, por lo que debemos especificar un código no utilizado de 0. Con legis.names especificamos los nombres de los votantes y con legis.dataespecificamos variables adicionales sobre nuestros votantes. Por último, desc y source nos permiten registrar información adicional sobre nuestros datos.

Con nuestros datos formateados correctamente, ahora podemos aplicar el modelo W-NOMINATE. el comando simplemente se llama wnominate :

resultado <-wnominate (rcObject = rc, polarity = c (1,1))

Con el argumento rcObject , simplemente nombramos nuestros datos formateados correctamente. El argumento de la polaridad , por el contrario, requiere una aportación sustancial del investigador: el usuario debe especificar un vector que enumere qué observación debe caer claramente en el lado positivo de cada dimensión estimada. Dada la política de la Guerra Fría, usamos la observación n. ° 1, EE. UU., Como ancla en ambas dimensiones que estimamos. Por defecto, wnominate coloca a los votantes en un espacio ideológico bidimensional (aunque esto podría cambiarse especificando la opción de atenuación ).

Para ver los resultados de nuestra estimación, podemos comenzar escribiendo: resumen (resultado) . Esto imprime la siguiente salida.

RESUMEN DEL OBJETO NOMINADO W

Número de legisladores: 59 (0 legisladores eliminados)

Número de votos: 219 (18 votos eliminados)

Número de dimensiones: 2

Sí pronosticado: 4693 de 5039 (93,1%) predicciones

correcto

Negativos previstos: 4125 de 4488 (91,9%) predicciones

correcto

Clasificación correcta: 89,5% 92,56%

APRE: 0,574 0,698

BPF: 0,783 0,841

Las primeras 10 estimaciones de los legisladores son:

coord1D coord2D

Estados Unidos 0,939 0,344

Canadá 0,932 0,362

Cuba 0,520 -0,385

Haití 0,362 -0,131

República Dominicana 0,796 -0,223

México 0,459 0,027

Guatemala 0,382 0,364

Honduras 0,588 -0,266

El Salvador 0,888 -0,460

Nicaragua 0,876 -0,301

La salida comienza recapitulando varias características descriptivas de nuestros datos y luego cambia para ajustar índices. Enumera, por ejemplo, la predicción correcta de sí y no, y luego, en “Clasificación correcta,” enumera el porcentaje predicho correctamente por la primera dimensión sola y luego ambas dimensiones juntas. Con un 89,5%, la primera dimensión puede explicar mucho por sí misma. El resultado finaliza enumerando las estimaciones de nuestras primeras 10 observaciones. Como puede verse, EE. UU. Tiene un valor positivo en ambas dimensiones, según nuestra especificación en el modelo.

Podemos obtener resultados adicionales escribiendo: plot (result) . El resultado de este comando se presenta en la Fig.  8.3 . El panel superior izquierdo visualiza los puntajes W-NOMINATE para las 59 naciones votantes. El eje horizontal es la primera dimensión de la puntuación, el eje vertical es la segunda dimensión y cada punto representa la posición de una nación. Por lo tanto, en el espacio ideológico bidimensional definido internamente en los procedimientos de la ONU, aquí es donde cae cada nación. El panel de la parte inferior izquierda muestra un gráfico de pantalla, que enumera el valor propio asociado con cada dimensión. Los valores propios más grandes indican que una dimensión tiene más poder explicativo. Como en todas las gráficas de pedregal, cada dimensión adicional tiene un valor explicativo menor que la anterior. 10El panel superior derecho muestra la distribución de los ángulos de las líneas de corte. Las líneas de corte dividen el sí de los votos negativos sobre un tema determinado. El hecho de que tantas líneas de corte estén cerca de la marca de 90 ∘ indica que la primera dimensión es importante para muchos de los votos. Finalmente, el panel inferior derecho muestra la malla de Coombs de este modelo, una visualización de cómo todas las líneas de corte de los 237 votos se unen en un solo espacio. Abrir imagen en nueva ventanaFigura 8.3 Figura 8.3 Gráfico de salida de la estimación de las puntuaciones de W-NOMINATE de las tres primeras sesiones de las Naciones Unidas

Si el usuario está satisfecho con los resultados de este modelo de medición, entonces es sencillo escribir las puntuaciones en un formato de datos utilizable. Dentro de nuestra wnominate llamado de salida resultado que podemos llamar el atributo llamado a los legisladores , lo que ahorra los puntos ideales para todos los países, las variables de llamada no rollo que hemos especificado (por ejemplo, Pacto de Varsovia o no), y una variedad de otras medidas. Guardamos esto como un nuevo marco de datos llamado puntuaciones y luego lo escribimos en un archivo CSV:

puntuaciones <-resultado $ legisladores

write.csv (puntuaciones, “UNscores.csv”)

Solo recuerde usar el comando setwd para especificar la carpeta en la que desea guardar el archivo de salida.

Una vez que tenemos nuestros puntos ideales W-NOMINAR en un marco de datos separado, podemos hacer cualquier cosa que haríamos normalmente con los datos en R , como dibujar nuestros propios gráficos. Supongamos que quisiéramos reproducir nuestro propio gráfico de los puntos ideales, pero queríamos marcar qué naciones eran miembros del Pacto de Varsovia frente a las que no lo eran. Podríamos hacer esto fácilmente usando nuestros datos de puntajes . La forma más fácil de hacer esto podría ser usar el comando subconjunto para crear marcos de datos separados de nuestros dos grupos:

wp.scores <-subconjunto (puntuaciones, V1 == “WP”)

other.scores <-subconjunto (puntuaciones, V1 == “Otro”)

Una vez que tengamos estos subconjuntos en la mano, podemos crear el gráfico relevante en tres líneas de código.

plot (x = other.scores $ coord1D, y = other.scores $ coord2D,

xlab = "Primera dimensión", ylab = "Segunda dimensión",  
  
 xlim = c (-1,1), ylim = c (-1,1), asp = 1)

puntos (x = wp.scores $ coord1D, y = wp.scores $ coord2D,

pch = 3, col = 'rojo')

leyenda (x = -1, y = -. 75, leyenda = c (“Otro,” “Pacto de Varsovia”),

col = c ("negro", "rojo"), pch = c (1,3))

En el llamado a la parcela , graficamos las 53 naciones que no eran miembros del Pacto de Varsovia, colocando la primera dimensión en el eje horizontal y la segunda en el eje vertical. Etiquetamos nuestros ejes apropiadamente usando xlab e ylab . También establecemos que los límites de nuestro gráfico van de -1 a 1 en ambas dimensiones, ya que los puntajes están restringidos a caer en estos rangos. Es importante destacar que, garantizamos que la escala de las dos dimensiones es el mismo, ya que en general debe para este tipo de modelo de medición, mediante el establecimiento de la asp relación ect a 1 ( asp = 1 ). En la segunda línea de código, usamos los puntoscomando para agregar las seis observaciones que estaban en el Pacto de Varsovia, coloreando estas observaciones de rojo y usando un carácter de trama diferente. Por último, agregamos una leyenda.

La figura 8.4 presenta la salida de nuestro código. Este gráfico transmite inmediatamente que la primera dimensión está capturando la división de la Guerra Fría entre los EE. UU. Y sus aliados versus la Unión Soviética y sus aliados. Especificamos que EE. UU. Tomaría coordenadas positivas en ambas dimensiones, por lo que podemos ver que los aliados soviéticos (representados con cruces rojas) están en los extremos de los valores negativos en la primera dimensión. Abrir imagen en nueva ventanaFigura 8.4 Figura 8.4 Gráfico de la primera y segunda dimensiones de las puntuaciones de W-NOMINATE de las tres primeras sesiones de las Naciones Unidas. Una cruz roja indica un miembro del Pacto de Varsovia y un círculo negro indica todos los demás miembros de la ONU (figura en color en línea)

En resumen, este capítulo ha ilustrado cuatro ejemplos de cómo usar paquetes R para implementar métodos avanzados. El hecho de que estos paquetes estén disponibles gratuitamente hace que el trabajo de vanguardia en metodología política y de una variedad de disciplinas esté fácilmente disponible para cualquier usuario de R. Ningún libro podría mostrar todos los paquetes aportados por los investigadores que están disponibles, sobre todo porque los nuevos paquetes están disponibles de forma regular. La próxima vez que se enfrente a un problema metodológico exigente, es posible que desee comprobar los servidores de CRAN para ver si alguien ya ha escrito un programa que proporcione lo que necesita.

8.5 Problemas de práctica Este conjunto de problemas de práctica considera cada una de las bibliotecas de ejemplo por turno, y luego sugiere que intente usar un paquete nuevo que no se ha discutido en este capítulo. Cada pregunta requiere un conjunto de datos único. 1. Regresión logística multinivel: revise Singh’s (2015) datos sobre la participación electoral en función de las reglas de votación obligatorias y varios otros predictores. Si no tiene el archivo stdSingh.dta , descárguelo del Dataverse (consulte la página vii) o del contenido del capítulo (consulte la página 125). (Estos datos se introdujeron por primera vez en la Sección  7.4 ) Vuelva a ajustar este modelo de regresión logística mediante el comando glmer e incluya una intersección aleatoria por país-año ( cntryyear ). Recuerde que el resultado es la participación ( votada ). La severidad de las reglas de votación obligatoria ( severidad ) interactúa con los primeros cinco predictores: edad ( edad ), conocimiento político ( polinfrel ), ingresos ( ingresos ), eficacia ( eficacia ) y partidismo ( partyID ). Se deben incluir cinco predictores más solo para efectos aditivos: magnitud del distrito ( dist\_magnitude ), número de partidos ( enep ), margen de victoria ( vicmarg\_dist ), sistema parlamentario ( parlamentario ) y PIB per cápita ( desarrollo ). Nuevamente, todas las variables predictoras se han estandarizado. ¿Qué aprende de este modelo de regresión logística multinivel estimado con glmer que no aprende de un modelo de regresión logística agrupado estimado con glm ?

1. Modelo Bayesiano de Poisson con MCMC: Determine cómo estimar un modelo de Poisson con MCMC usando MCMCpack . Recargar Peake y Eshbaugh-Soha’s (2008) datos sobre la cobertura de noticias sobre política energética, discutidos por última vez en la sec. 7.3 Si no tiene el archivo PESenergy.csv , puede descargarlo del Dataverse (ver página vii) o del contenido del capítulo (ver página 125). Estimar un modelo Bayesiano de Poisson en el que el resultado es la cobertura energética ( Energía ), y las entradas son seis indicadores para los discursos presidenciales ( rmn1173 , grf0175 , grf575 , jec477 , jec1177 y jec479 ), un indicador del embargo petrolero árabe ( embargo ). , un indicador de la crisis de los rehenes de Irán ( rehenes ), el precio del petróleo ( oilc ), aprobación presidencial ( Aprobación ) y la tasa de desempleo ( Desempleo ). Utilice una prueba de Geweke para determinar si existe alguna evidencia de no convergencia. ¿Cómo debería cambiar su código en R si la no convergencia es un problema? Resuma sus resultados en una tabla y muestre una gráfica de densidad del coeficiente parcial en el discurso de Richard Nixon de noviembre de 1973 ( rmn1173 ).
2. Coincidencia exacta tosca: en el cap. 5 , los problemas de práctica introducidos por Álvarez et al. ( 2013) datos de un experimento de campo en Salta, Argentina, en el que algunos votantes emitieron su voto a través del voto electrónico y otros votaron en el entorno tradicional. Cargue la biblioteca externa y abra los datos en formato Stata. Si no tiene el archivo alpl2013.dta , puede descargarlo del Dataverse (consulte la página vii) o del contenido del capítulo (consulte la página 125). En este ejemplo, la variable de tratamiento es si el votante utilizó el voto electrónico o el voto tradicional ( EV ). Las covariables son grupo de edad ( age\_group ), educación ( educ ), trabajador de cuello blanco ( white\_collar ), no un trabajador de tiempo completo ( not\_full\_time ), hombre ( hombre), una variable de conteo para el número de seis posibles dispositivos tecnológicos utilizados ( tecnología ) y una escala ordinal para el conocimiento político ( pol\_info ). Utilice la biblioteca cem para responder lo siguiente: un. ¿Cuán equilibradas están las observaciones de tratamiento y control en los datos brutos?
3. Realice una coincidencia exacta aproximada con el comando cem . ¿Cuánto ha mejorado el equilibrio como resultado?
4. Considere tres posibles variables de respuesta: si el votante evaluó positivamente la experiencia de votación ( eval\_voting ), si el votante evaluó la velocidad de la votación como rápida ( velocidad ) y si el votante está seguro de que su voto está siendo contado ( sure\_counted ). ¿Cuál es el efecto del tratamiento promedio en los tratados (ATT) en su conjunto de datos emparejados para cada una de estas tres respuestas?
5. ¿En qué se diferencian sus estimaciones de los efectos promedio del tratamiento en los tratados de las pruebas simples de diferencia de medias?
6. W-NOMINATE: De vuelta en la Secta. 2.1 , presentamos los datos del pase de lista de Lewis y Poole para el 113º Senado de los Estados Unidos. Consulte el código allí para leer estos datos, que están en formato de ancho fijo. El nombre del archivo es sen113kh.ord y está disponible en Dataverse (consulte la página vii) y el contenido del capítulo (consulte la página 125).  
   un. Formatee los datos como una matriz y cree lo siguiente: una matriz separada solo de los 657 pases de lista, un vector de los números de identificación de ICPSR y una matriz de las variables que no pasan de lista. Utilice todos estos para crear un objeto de clase rollcall . Las votaciones nominales se codifican de la siguiente manera: 1 = Sí, 6 = No, 7 y 9 = falta y 0 = no es miembro.
7. Estime un modelo W-NOMINATE bidimensional para este objeto de lista. Del resumen de sus resultados, informe lo siguiente: ¿Cuántos legisladores fueron eliminados? ¿Cuántos votos se eliminaron? ¿Fue la clasificación general correcta?
8. Examine la gráfica de salida de su modelo estimado, incluidas las coordenadas W-NOMINATE y la gráfica de la pantalla. Con base en el diagrama de pantalla, ¿cuántas dimensiones cree que son suficientes para caracterizar el comportamiento de votación en el 113 ° Senado? ¿Por qué?
9. Bonificación: intente aprender a usar un paquete que nunca antes haya usado. Instale el paquete Amelia , que realiza una imputación múltiple de los datos faltantes. Eche un vistazo a Honaker et al. (2011) Artículo en el Journal of Statistical Software para tener una idea de la lógica de imputación múltiple y para aprender cómo hacer esto en R . Ajuste un modelo lineal en conjuntos de datos imputados utilizando los datos de comercio libre de la biblioteca Amelia . ¿Qué encuentras?

Notas al pie 1 . Tenga en cuenta que este enfoque multinivel de datos de panel es más sensato para paneles cortos como estos, donde hay muchos individuos en relación con el número de puntos de tiempo. Para paneles largos en los que hay muchos puntos de tiempo en relación con el número de individuos, los modelos más apropiados se describen como métodos de sección transversal de series de tiempo agrupadas. Para obtener más información sobre el estudio de paneles cortos, consulte Monogan (2011) y Fitzmaurice et al. (2004).

2 . Si no tiene estos datos de antes, puede descargar el archivo BPchap7.dta del Dataverse en la página vii o el contenido del capítulo en la página 125.

3 . Consulte también la biblioteca nlme , que fue un predecesor de lme4 .

4 . Si no tiene estos datos de antes, puede descargar el archivo SinghJTP.dta del Dataverse en la página vii o el contenido del capítulo en la página 125.

5 . Para los priores de los coeficientes, la opción b0 establece el vector de medias de un anterior gaussiano multivariado y B0 establece la matriz de varianza-covarianza del anterior gaussiano multivariante. La distribución previa de la varianza del error de la regresión es Gamma inversa, y esta distribución puede manipularse estableciendo su parámetro de forma con la opción c0 y el parámetro de escala con la opción d0 . Alternativamente, la distribución Gamma inversa se puede manipular cambiando su media con la opción sigma.mu y su varianza con la opción sigma.var .

6 . Para escribir una tabla similar a la Tabla 8.2 en LaTeX, cargue la biblioteca xtable en R y escriba lo siguiente en la consola:

xtable (cbind (resumen (mcmc.horas) $ estadísticas [, 1: 2], resumen (mcmc.horas) $ cuantiles [, c (1,5)]), dígitos = 4)

7 . Esto ocurre con frecuencia cuando un paquete depende del código de otro.

8 . Los datos de LaLonde también están disponibles en el archivo LL.csv , disponible en el Dataverse (ver página vii) o el contenido del capítulo (ver página 125).

9 . Los datos de la ONU también están disponibles en el archivo UN.csv , disponible en el Dataverse (ver página vii) o el contenido del capítulo (ver página 125).

10 . Al elegir cuántas dimensiones incluir en un modelo de medición, muchos académicos usan la “regla del codo,” lo que significa que no incluyen ninguna dimensión más allá de un codo visual en el diagrama de la pantalla. En este caso, un erudito ciertamente no incluiría más de tres dimensiones y podría contentarse con dos. Otro límite común es incluir cualquier dimensión para la cual el valor propio exceda 1, lo que nos haría detenernos en dos dimensiones.

Material suplementario 318886\_1\_En\_8\_MOESM1\_ESM.zip (2.5 mb) Dataverse (2,154 KB) Referencias Alvarez RM, Levin I, Pomares J, Leiras M (2013) Votar de forma segura y sencilla: el impacto del voto electrónico en la percepción ciudadana. Polit Sci Res Methods 1 (1): 117-137 CrossRefGoogle Académico Bates D, Maechler M, Bolker B, Walker S (2014) lme4 : modelos lineales de efectos mixtos que utilizan Eigen y S4. Versión del paquete R 1.1-7. <http://www.CRAN.R-project.org/package=lme4> Berkman M, Plutzer E (2010) Evolución, creacionismo y la batalla por controlar las aulas de Estados Unidos. Cambridge University Press, Nueva York CrossRefGoogle Académico Black D (1958) La teoría de los comités y las elecciones. Cambridge University Press, Londres zbMATHGoogle Académico Carlin BP, Louis TA (2009) Métodos bayesianos para el análisis de datos. Chapman & Hall / CRC, Boca Ratón, FL zbMATHGoogle Académico Downs A (1957) Una teoría económica de la democracia. Harper and Row, Nueva York Google Académico Fitzmaurice GM, Laird NM, Ware JH (2004) Análisis longitudinal aplicado. Wiley-Interscience, Hoboken, Nueva Jersey zbMATHGoogle Académico Gelman A, Hill J (2007) Análisis de datos utilizando modelos de regresión y multinivel / jerárquicos. Cambridge University Press, Nueva York Google Académico Gelman A, Carlin JB, Stern HS, Rubin DB (2004) Análisis de datos bayesianos, 2ª ed. Chapman & Hall / CRC, Boca Ratón, FL zbMATHGoogle Académico Gill J (2008) Métodos bayesianos: un enfoque de las ciencias sociales y del comportamiento, 2ª ed. Chapman & Hall / CRC, Boca Ratón, FL zbMATHGoogle Académico Honaker J, King G, Blackwell M (2011) Amelia II: un programa para datos faltantes. J Stat Softw 45 (7): 1–47 CrossRefGoogle Académico Hotelling H (1929) Estabilidad en competencia. Econ J 39 (153): 41–57 CrossRefGoogle Académico Iacus SM, King G, Porro G (2009) cem : software para emparejamiento exacto aproximado. J Stat Softw 30 (9): 1–27 CrossRefGoogle Académico Iacus SM, King G, Porro G (2011) Métodos de emparejamiento multivariante que limitan el desequilibrio monótono. J Am Stat Assoc 106 (493): 345–361 CrossRefMathSciNetzbMATHGoogle Académico Iacus SM, King G, Porro G (2012) Inferencia causal sin verificación de saldo: coincidencia exacta aproximada. Polit Anal 20 (1): 1–24 CrossRefGoogle Académico Imai K, van Dyk DA (2004) Inferencia causal con regímenes de tratamiento general: generalización de la puntuación de propensión. J Am Stat Assoc 99 (467): 854–866 CrossRefMathSciNetzbMATHGoogle Académico Laird NM, Fitzmaurice GM (2013) Modelado de datos longitudinal. En: Scott MA, Simonoff JS, Marx BD (eds) El manual Sage de modelado multinivel. Sage, Thousand Oaks, CA Google Académico LaLonde RJ (1986) Evaluación de las evaluaciones econométricas de programas de formación con datos experimentales. Am Econ Rev 76 (4): 604–620 Google Académico Martin AD, Quinn KM, Park JH (2011) MCMCpack : Markov chain Monte Carlo en R. J Stat Softw 42 (9): 1–21 CrossRefGoogle Académico McCarty NM, Poole KT, Rosenthal H (1997) Redistribución de la renta y realineamiento de la política estadounidense. Estudios del instituto empresarial estadounidense sobre la comprensión de la desigualdad económica. AEI Press, Washington, DC Google Académico Monogan JE III (2011) Análisis de datos de panel. En: Badie B, Berg-Schlosser D, Morlino L (eds) Enciclopedia internacional de ciencia política. Sage, Thousand Oaks, CA Google Académico Peake JS, Eshbaugh-Soha M (2008) El impacto en el establecimiento de la agenda de los principales discursos televisivos presidenciales. Polit Commun 25: 113-137 CrossRefGoogle Académico Poole KT, Rosenthal H (1997) Congreso: una historia político-económica de la votación nominal. Oxford University Press, Nueva York Google Académico Poole KT, Lewis J, Lo J, Carroll R (2011) Escala de votos nominales con wnominate en R. J Stat Softw 42 (14): 1–21 CrossRefGoogle Académico Robert CP (2001) La elección bayesiana: de los fundamentos de la teoría de la decisión a la implementación computacional, 2ª ed. Springer, Nueva York zbMATHGoogle Académico Rubin DB (2006) Muestreo emparejado para efectos causales. Cambridge University Press, Nueva York CrossRefzbMATHGoogle Académico Scott MA, Simonoff JS, Marx BD (eds) (2013) El manual Sage de modelado multinivel. Sage, Thousand Oaks, CA Google Académico Sekhon JS, Grieve RD (2012) Un método de emparejamiento para mejorar el equilibrio de covariables en los análisis de rentabilidad. Health Econ 21 (6): 695–714 CrossRefGoogle Académico Singh SP (2014a) Funciones de pérdida de utilidad lineales y cuadráticas en la investigación del comportamiento electoral. J Theor Polit 26 (1): 35–58 CrossRefGoogle Académico Singh SP (2015) Voto obligatorio y cálculo de decisiones de participación. Polit Stud 63 (3): 548–568 CrossRefGoogle Académico

# 9 Análisis de series temporales

Palabras clave: - Series de tiempo - Mínimos cuadrados ordinarios - Análisis de series temporales - Correlación en serie - Variable endógena

La mayoría de los métodos descritos hasta ahora en este libro están orientados principalmente al análisis transversal, o al estudio de una muestra de datos tomados en el mismo momento. En este capítulo, pasamos a los métodos para modelar una serie de tiempo, o una variable que se observa secuencialmente a intervalos regulares a lo largo del tiempo (por ejemplo, diaria, semanal, mensual, trimestral o anual). Los datos de series de tiempo con frecuencia tienen tendencias y procesos de error complejos, por lo que no tener en cuenta estas características puede producir resultados falsos (Granger y Newbold 1974). Han surgido varios enfoques para el análisis de series de tiempo para abordar estos problemas y prevenir inferencias falsas. Dentro de la ciencia política, los académicos de la opinión pública, la economía política, los conflictos internacionales y varios otros temas trabajan regularmente con datos referenciados en el tiempo, por lo que las herramientas adecuadas para el análisis de series de tiempo son importantes en el análisis político.

Muchos investigadores no piensan en R como un programa para el análisis de series de tiempo, sino que utilizan software especializado como Autobox, EViews o RATS. Incluso SAS y Stata tienden a conseguir más atención para el análisis de series temporales de R hace. Sin embargo, R en realidad tiene una amplia gama de comandos para modelos de series de tiempo, particularmente a través de los paquetes TSA y vars . En este capítulo, ilustraremos tres enfoques para el análisis de series de tiempo en R : el enfoque de Box-Jenkins, extensiones a modelos lineales estimados con mínimos cuadrados y autorregresión vectorial. Esta no es una lista exhaustiva de las herramientas disponibles para estudiar series de tiempo, pero solo pretende presentar algunos métodos destacados. Ver secc. 9.4 para leer más sobre series de tiempo.

Tanto el enfoque de Box-Jenkins como las extensiones de los modelos lineales son ejemplos de modelos de series de tiempo de ecuación única. Ambos enfoques tratan una serie de tiempo como una variable de resultado y se ajustan a un modelo de ese resultado que puede expresarse en una ecuación, al igual que los modelos de regresión de los capítulos anteriores pueden expresarse en una sola ecuación. Dado que ambos enfoques entran en esta categoría amplia, el conjunto de datos de trabajo que usamos tanto para el enfoque de Box-Jenkins como para las extensiones de los modelos lineales será el de Peake y Eshbaugh-Soha (2008) datos mensuales sobre la cobertura televisiva de la política energética que se introdujeron por primera vez en el Cap. 3 Por el contrario, la autorregresión vectorial es un modelo de series de tiempo de ecuaciones múltiples (para más detalles sobre esta distinción, consulte Brandt y Williams 2007 o Lütkepohl 2005). Con un modelo de autorregresión vectorial, dos o más series de tiempo se consideran endógenas, por lo que se requieren múltiples ecuaciones para especificar completamente el modelo. Esto es importante porque las variables endógenas pueden afectarse entre sí y, para interpretar el efecto de una variable de entrada, se debe considerar el contexto más amplio del sistema completo. Dado que los modelos de ecuaciones múltiples tienen una especificación tan diferente, al discutir la autorregresión vectorial, el ejemplo de trabajo será Brandt y Freeman (2006) análisis de acciones políticas semanales en el conflicto israelo-palestino; Se plantearán más detalles una vez que lleguemos a esa sección.

9.1 El método Box-Jenkins El enfoque de Box-Jenkins para las series de tiempo se basa en modelos de promedio móvil integrado autorregresivo (ARIMA) para capturar el proceso de error en los datos. Para obtener una explicación completa de este enfoque, consulte Box et al. (2008). La lógica básica de este enfoque es que una serie de tiempo debe filtrarse, o blanquearse previamente , de cualquier tendencia y proceso de error antes de intentar ajustar un modelo inferencial. Una vez que se ha especificado un modelo del ruido o error, el investigador puede proceder a probar las hipótesis. 1 Debido a las formas funcionales no lineales que a menudo surgen en ARIMA y en los modelos de función de transferencia, estas generalmente se estiman con la máxima probabilidad.

Para ilustrar el proceso desde la identificación de un modelo de error hasta la prueba de una hipótesis, revisamos la de Peake y Eshbaugh-Soha (2008) Serie temporal mensual sobre cobertura de pólizas energéticas. Comenzamos recargando nuestros datos: 2

pres.energy <-read.csv (“PESenergy.csv”)

Nuestra variable de resultado es el número de historias relacionadas con la energía en los noticieros de televisión nocturnos por mes ( Energía ). Un buen primer paso es trazar la serie que se está estudiando para ver si alguna tendencia u otras características son evidentes de inmediato. Consulte la figura  3.6 para ver el código de ejemplo que usamos para trazar nuestra variable de resultado ( Energía ) y el precio del petróleo, que es uno de los predictores ( petróleoc ). Como un primer vistazo a la serie de cobertura de noticias, los datos parecen estar estacionarios , lo que significa que se mueven alrededor de una media sin tener una tendencia en una dirección o mostrar un patrón integrado .

Como punto sustancialmente importante sobre las series de tiempo en general, la distinción entre series estacionarias y no estacionarias es importante. Muchos métodos de series de tiempo están diseñados específicamente para modelar series estacionarias, por lo que su aplicación a una serie no estacionaria puede resultar problemático. Una serie estacionaria es aquella en la que la media y la varianza no cambian condicionalmente en el tiempo, y la serie no tiene una tendencia en una dirección. Si una serie estacionaria se altera o se mueve a un valor más alto o más bajo, eventualmente regresa a un nivel de equilibrio. Los dos tipos de series no estacionarias son series de tendencia y series integradas. Con tendenciaserie, como su nombre lo indica, el promedio condicional cambia con el tiempo, por lo general subiendo o bajando de manera constante. Por ejemplo, los precios nominales de los bienes de consumo pueden subir o bajar de un trimestre a otro, pero el valor medio tiende a subir con el tiempo. Una serie integrada, también llamada serie de raíz unitaria, no tiene un valor de equilibrio y tiene la característica de “memoria larga.” Esto significa que si algo cambia el valor de una serie, los valores futuros de la serie se verán afectados por ese cambio en el futuro. Por ejemplo, supongamos que el promedio industrial Dow Jones cayó 300 puntos en un día, de 18.200 a 17.900. En tal caso, los valores futuros del Dow se basarán en el nuevo valor de 17.900 más lo mucho que subió o bajó el valor cada día subsiguiente, y el Dow no tenderá a volver a los valores anteriores. Es importante destacar que al diferenciarel valor actual de una serie a partir de su valor anterior, o midiendo cuánto cambió una serie de una vez a la siguiente, las series integradas o en tendencia se pueden hacer estacionarias. Por lo tanto, se pueden aplicar muchos modelos para series estacionarias a la diferencia de una serie no estacionaria. Siempre que realice un trabajo de series de tiempo aplicadas, observe siempre el gráfico de la serie, como la figura  3.6 en este caso, así como los diagnósticos que se describirán para determinar si está estudiando una serie estacionaria, de tendencia o de raíz unitaria. 3

Como segundo paso, nos volvemos a medidas de diagnóstico y vistazo a la una uto c orrelation f unción (ACF) y p artial un uto c orrelation f unción (FAP). La función de autocorrelación muestra cuánto se correlacionan los valores actuales de una serie con los valores anteriores en un cierto retraso . Retrasosson importantes para el modelado de series de tiempo, como se verá más adelante en el capítulo, y actúan cambiando el índice de una serie para hacer retroceder el tiempo y formar una nueva serie. Por ejemplo, un retraso de primer orden es una nueva serie donde cada observación es el valor que ocurrió un período de tiempo anteriormente. Un retraso de segundo orden es una nueva serie en la que cada observación es el valor de dos períodos de tiempo anteriores, y así sucesivamente. Esto es importante cuando se usa la función de autocorrelación porque, si estamos estudiando una serie y con índice de tiempo t , ACF (1) nos daría la correlación entre y t y y t −1 . (Para la cobertura de energía, entonces, ACF (1) nos dice la correlación entre la cobertura de los meses actuales con la cobertura de los meses anteriores). Posteriormente, ACF (2) sería la autocorrelación entre y t y y t −2 , y más generalmente ACF (p) es la autocorrelación entre y t y y t - p . El PACF proporciona la autocorrelación entre los valores actuales y rezagados que no se tienen en cuenta en todos los rezagos anteriores . Entonces, para el primer retraso, ACF (1) = PACF (1), pero para todos los retrasos posteriores, solo la autocorrelación única de ese retraso aparece en el PACF. Si quisiéramos conocer los ACF y PACF de nuestra serie de políticas energéticas, escribiríamos:

acf (pres.energy $ Energy, lag.max = 24)

pacf (pres.energy $ Energy, lag.max = 24)

Las funciones acf y pacf están disponibles sin cargar un paquete, aunque el código cambia ligeramente si los usuarios cargan el paquete TSA . 4 Observe que dentro de las funciones acf y pacf , primero enumeramos las series que estamos diagnosticando. En segundo lugar, designamos lag.max , que es el número de retrasos de autocorrelación que deseamos considerar. Dado que estos son datos mensuales, 24 rezagos nos dan 2 años de rezagos. En algunas series, la estacionalidademergerá, en el que vemos evidencia de valores similares en la misma época cada año. Esto se vería con rezagos significativos alrededor de 12 y 24 con datos mensuales (o alrededor de 4, 8 y 12, por el contrario, con datos trimestrales). No aparece tal evidencia en este caso.

Los gráficos de nuestro ACF y PACF se muestran en la Fig.  9.1 . En cada una de estas figuras, el eje horizontal representa la longitud del rezago y el eje vertical representa el valor de correlación (o correlación parcial). En cada retraso, la autocorrelación para ese retraso se muestra con una línea sólida similar a un histograma desde cero hasta el valor de la correlación. Las líneas discontinuas azules representan el umbral para una correlación significativa. Específicamente, las bandas azules representan el intervalo de confianza del 95% basado en una serie no correlacionada. 5 Todo esto significa que si la línea similar a un histograma no cruza la línea discontinua, el nivel de autocorrelación no es perceptiblemente diferente de cero, pero los picos de correlación fuera de la banda de error son estadísticamente significativos. Abrir imagen en nueva ventanaFigura 9.1 Figura 9.1 Función de autocorrelación y función de autocorrelación parcial de la cobertura televisiva mensual de la política energética a través de 24 rezagos. ( a ) ACF. ( b ) PACF

El ACF, en la figura  9.1a comienza mostrando la autocorrelación de retardo cero, que es cuánto se correlacionan los valores actuales con ellos mismos, siempre exactamente 1.0. Luego, vemos los valores más informativos de autocorrelación en cada rezago. Con un retraso, por ejemplo, la correlación en serie es 0,823. Para dos rezagos, cae a 0.682. El sexto retraso es el último retraso en mostrar una autocorrelación discernible, por lo que podemos decir que el ACF decae rápidamente. El PACF, en la figura  9.1 b, omite el retardo cero y comienza con una correlación serial de primer orden. Como se esperaba, esto es 0.823, tal como mostró el ACF. Sin embargo, una vez que tenemos en cuenta la correlación serial de primer orden, los términos de autocorrelación parcial en los retrasos posteriores no son estadísticamente discernibles. 6

En este punto, determinamos qué proceso de error ARIMA dejaría una huella empírica como la que muestran este ACF y PACF. Para obtener más detalles sobre las huellas comunes, consulte Enders (2009, pag. 68). Notacionalmente, llamamos al proceso de error ARIMA ( p , d , q ), donde p es el número de términos autorregresivos, d es la cantidad de veces que la serie necesita ser diferenciada y q es el número de términos de promedio móvil. Funcionalmente, un modelo ARMA general , que incluye componentes de media móvil y autorregresiva, se escribe de la siguiente manera: yt=α0+∑i = 1pagαIyt - yo+∑i = 1qβIϵt - yo+ϵt (9,1) Aquí, y t es nuestra serie de interés, y este modelo ARMA se convierte en un modelo ARIMA cuando decidimos si necesitamos diferenciar y t o no. Observe que y t se retrasa p veces para los términos autorregresivos, y el término de perturbación ( ε t ) se retrasa q veces para los términos de promedio móvil. En el caso de la cobertura de la póliza energética, el ACF muestra un rápido decaimiento y vemos un pico significativo en el PACF, por lo que podemos decir que estamos ante un proceso autorregresivo de primer orden, denotado AR (1) o ARIMA (1,0 , 0). Una vez que hemos identificado nuestra una uto r egressive i NTEGRADO m Oving un modelo verage, se puede estimar utilizando la Arima función:

ar1.mod <-arima (pres.energy $ Energy, order = c (1,0,0))

La primera entrada es la serie que estamos modelando, y la opción de orden nos permite especificar p , d y q (en orden) para nuestro proceso ARIMA ( p , d , q ). Al escribir ar1.mod , vemos la salida de nuestros resultados:

Llamada:

arima (x = energía pres. $ Energía, orden = c (1, 0, 0))

Coeficientes:

intercepción ar1  
  
 0.8235 32.9020

se 0.0416 9.2403

sigma ^ 2 estimado como 502,7: probabilidad logarítmica = -815,77,

aic = 1637.55

Simplemente, esto nos muestra la estimación y el error estándar del coeficiente autorregresivo ( ar1 ) y la intersección, así como la varianza residual, la probabilidad logarítmica y el AIC.

El siguiente paso en el proceso de modelado de Box-Jenkins es diagnosticar si el modelo estimado filtra suficientemente los datos. 7 Hacemos esto de dos maneras: Primero, estudiando el ACF y PACF para los residuos del modelo ARIMA. El código es:

acf (ar1.mod $ residuales, lag.max = 24)

pacf (ar1.mod $ residuales, lag.max = 24)

Al igual que con muchos otros modelos, podemos llamar a nuestros residuos usando el nombre del modelo y un signo de dólar ( ar1.mod $ residuales ). Los gráficos resultantes se presentan en la figura  9.2 . Como muestran tanto el ACF como el PACF, el segundo y el cuarto rezagos apenas cruzan el umbral de significancia, pero no hay un patrón claro o evidencia de una característica pasada por alto del proceso de error. La mayoría (pero no todos) los analistas estarían satisfechos con este patrón en estas cifras. Abrir imagen en nueva ventanaFigura 9.2 Figura 9.2 Función de autocorrelación y función de autocorrelación parcial para los residuos del modelo AR (1) a través de 24 rezagos. Las líneas discontinuas azules representan un intervalo de confianza del 95% para una serie no correlacionada. ( a ) ACF. ( b ) PACF (figura de color en línea)

Como segundo paso de diagnosticar si hemos filtrado suficientemente los datos, se calcula el Ljung- Box Q - prueba . Esta es una prueba conjunta de varios rezagos para determinar si existe evidencia de correlación serial en alguno de los rezagos. La hipótesis nula es que los datos son independientes, por lo que un resultado significativo sirve como evidencia de un problema. La sintaxis de esta prueba es:

Box.test (ar1.mod $ residuals, lag = 24, type = “Ljung-Box”)

Primero especificamos la serie de interés, luego, con la opción de retraso, indicamos cuántos retrasos deben entrar en la prueba y, por último, especificamos con el tipo Ljung-Box (a diferencia de la prueba Box-Pierce, que no funciona tan bien). Nuestros resultados de esta prueba son:

Prueba de Box-Ljung

datos: ar1.mod $ residuales

X al cuadrado = 20,1121, gl = 24, valor p =

0,6904

El estadístico de nuestra prueba no es significativo ( p  = 0. 6904), por lo que esta prueba no muestra evidencia de correlación serial durante 2 años de retraso. Si estamos satisfechos de que AR (1) caracteriza nuestro proceso de error, podemos proceder al modelado real en la siguiente sección.

9.1.1 Funciones de transferencia frente a modelos estáticos Para estimar un modelo teóricamente motivado, necesitamos establecer una distinción importante entre dos tipos de formas funcionales: por un lado, una forma funcional estática supone que los valores actuales de los predictores afectan los valores actuales de los resultados. Por ejemplo, podemos creer que el precio del petróleo en un mes determinado afecta la cobertura de temas energéticos en la televisión en el mismo mes, mientras que los precios del petróleo del próximo mes afectarán la cobertura energética del próximo mes. Si pensamos que este es el único proceso en curso, entonces queremos una forma funcional estática. Por otro lado, podemos utilizar una dinámicaforma funcional. En el caso de los precios del petróleo, esto significaría que los precios del petróleo de este mes afectarían la cobertura energética de este mes, y también la cobertura del mes siguiente en menor grado, el mes siguiente en menor grado aún, y así sucesivamente. Esencialmente, una forma funcional dinámica permite efectos secundarios de cada observación de una entrada.

Estimación de un modelo de regresión estática con un proceso de error ARIMA sólo requiere dos pasos en R . Primero, todas las variables predictoras deben colocarse en una matriz. En segundo lugar, simplemente podemos agregar una opción a nuestra función arima :

predictores <-as.matrix (subconjunto (pres.energy, select = c (rmn1173,

grf0175, grf575, jec477, jec1177, jec479, embargo, rehenes,  
  
 oilc, Aprobación, Desempleo)))

static.mod <-arima (pres.energy $ Energy, order = c (1,0,0),

xreg = predictores)

En este código, primero subconjuntamos nuestro conjunto de datos original y tratamos el subconjunto como una matriz denominada predictores . En segundo lugar, usamos la misma función arima que usamos para estimar nuestro modelo de ruido AR (1), pero agregamos la opción xreg = predictores . Esto ahora estima un modelo en el que se corrige el proceso de error temporal, pero también incluimos predictores de interés motivados teóricamente. Si escribimos static.mod , la salida es:

Llamada:

arima (x = energía pres. $ Energía, orden = c (1,0,0), xreg =

predictores)

Coeficientes:

ar1 intercepción rmn1173 grf0175 grf575  
  
 0,8222 5,8822 91,3265 31,8761 -8,2280

se 0.0481 52.9008 15.0884 15.4643 15.2025

jec477 jec1177 jec479 embargo rehenes  
  
 29.6446 -6.6967 -20.1624 35.3247 -16.5001

se 15.0831 15.0844 15.2238 15.1200 13.7619

Oilc Aprobación Desempleo  
  
 0,8855 -0,2479 1,0080

se 1.0192 0.2816 3.8909

sigma ^ 2 estimado como 379,3: probabilidad logarítmica = -790,42,

aic = 1608,84

Esto ahora nos muestra la estimación y el error estándar no solo para el coeficiente autorregresivo y la intersección, sino también para el coeficiente de regresión parcial de todos los demás predictores del modelo. Nuevamente, las tres medidas de ajuste se informan al final de la impresión.

Si quisiéramos incluir uno o más predictores que tengan un efecto dinámico, entonces recurrimos al método de las funciones de transferencia , que especifican que un predictor tiene un efecto de desbordamiento en las observaciones posteriores. Un caso especial de esto se llama análisis de intervención , en el que un tratamiento se codifica con un indicador (Box y Tiao 1975). Con el análisis de intervención, pueden surgir una variedad de formas funcionales, dependiendo en gran medida de si la variable indicadora está codificada como un pulso (tomando un valor de 1 solo en el momento del tratamiento y 0 en caso contrario) o un paso (tomando un valor de 0 antes del tratamiento). el tratamiento y 1 en todos los momentos posteriores). Es recomendable probar ambas codificaciones y leer más en la forma funcional de cada una (Enders 2009, Secta. 5.1 ).

Como ilustración, ajustamos una función de transferencia para una intervención de pulso para el discurso de Richard Nixon en noviembre de 1973. Para estimar este modelo, ahora necesitamos cargar el paquete TSA para acceder a la función arimax ( ARIMA con predictores x ). Recuerde que es posible que deba usar install.packages para descargar TSA :

install.packages (“TSA”)

biblioteca (TSA)

dynamic.mod <-arimax (pres.energy $ Energy, order = c (1,0,0),

xreg = predictores [, - 1], xtransf = predictores [, 1],  
  
 transferencia = lista (c (1,0)))

La sintaxis de arimax es similar a la de arima , pero ahora se nos permiten algunas opciones más para las funciones de transferencia. Observe en este caso que usamos el código xreg = predictores [-1] para eliminar el indicador del discurso de noviembre de 1973 de Nixon de los predictores estáticos. En su lugar, colocamos este predictor con la opción xtransf . Lo último que debemos hacer es especificar el orden de nuestra función de transferencia, lo que hacemos con la opción de transferencia . La opción de transferencia acepta una lista de vectores, un vector por predictor de función de transferencia. Para nuestra única función de transferencia, especificamos c (1,0): El primer término se refiere al orden del término de decaimiento dinámico (por lo que un 0 aquí en realidad vuelve a ser un modelo estático), y el segundo término se refiere a la duración del retraso del efecto del predictor (por lo que si esperábamos que un efecto creciera , podríamos poner un número mayor que 0 aquí). Con estos ajustes, decimos que el discurso de Nixon tuvo un efecto en el mes que lo pronunció, y luego el efecto se extendió a los meses siguientes a un ritmo decreciente.

Al escribir dynamic.mod , obtenemos nuestro resultado:

Llamada:

arimax (x = energía pres. $ Energía, orden = c (1,0,0),

xreg = predictores [, - 1],

xtransf = predictores [, 1], transferencia = lista (c (1, 0)))

Coeficientes:

ar1 intercepción grf0175 grf575 jec477  
  
 0,8262 20,2787 31,5282 -7,9725 29,9820

se 0.0476 46.6870 13.8530 13.6104 13.5013

jec1177 jec479 embargo rehenes oilc  
  
 -6,3304 -19,8179 25,9388 -16,9015 0,5927

se 13.5011 13.6345 13.2305 12.4422 0.9205

Aprobación Desempleo T1-AR1 T1-MA0  
  
 -0,2074 0,1660 0,6087 160,6241

se 0,2495 3,5472 0,0230 17,0388

sigma ^ 2 estimado como 305,1: probabilidad logarítmica = -770,83,

aic = 1569,66

El resultado es similar al del modelo de regresión ARIMA estático, pero ahora hay dos términos para el efecto del habla de Nixon. El primero, T1-AR1 , da el término de desintegración. Cuanto más cercano esté este término a 1, más persistente será el efecto de la variable. El segundo término, T1-MA0 , es el efecto inicial del discurso sobre la cobertura energética en el mes en que se dio. 8En términos de ajuste del modelo, observe que la salida de cada ARIMA o modelo de función de transferencia que hemos estimado informa el criterio de información de Akaike (AIC). Con esta medida, las puntuaciones más bajas indican un mejor ajuste penalizado. Con una puntuación de 1569,66, este modelo de función de transferencia dinámica tiene el AIC más bajo de cualquier modelo que hayamos ajustado a estos datos. Por lo tanto, el modelo dinámico tiene un mejor ajuste que el modelo estático o el modelo ateórico AR (1) sin predictores.

Sin embargo, para tener una idea real del efecto de un análisis de intervención, un analista siempre debe intentar dibujar el efecto que modeló. (Nuevamente, es clave estudiar la forma funcional detrás de la especificación de intervención elegida, como lo describe Enders 2009, Secta. 5.1 .) Para dibujar el efecto de nuestra intervención para el discurso de Nixon de 1973, escribimos:

meses <-c (1: 180)

y.pred <-dynamic.mod $ coef [2:12]% \*% c (1, predictores [58, -1]) +

160,6241 \* predictores [, 1] +  
  
 160.6241 \* (. 6087 ^ (meses-59)) \* como numérico (meses> 59)

plot (y = energía pres. $ Energía, x = meses, xlab = “Mes,”

ylab = "Historias de políticas energéticas", tipo = "l", ejes = F)

eje (1, en = c (1,37,73,109,145), etiquetas = c (“Enero de 1969,”

"Ene. 1972", "Ene. 1975", "Ene. 1978", "Ene. 1981"))

eje (2)

caja()

líneas (y = y.pred, x = meses, lty = 2, col = “blue,” lwd = 2)

En la primera línea, simplemente creamos un índice de tiempo para los 180 meses del estudio. En la segunda línea, creamos valores predichos para el efecto de la intervención manteniendo todo lo demás igual. Una suposición crítica que hacemos es que mantenemos todos los demás predictores iguales al establecerlos en sus valores de octubre de 1973, el mes 58 de la serie (por lo tanto, predictores [58, -1]). Entonces, considerando los componentes de esta segunda línea, el primer término multiplica los coeficientes de los predictores estáticos por sus últimos valores antes de la intervención, el segundo término captura el efecto de la intervención en el mes del discurso y el tercer término captura el efecto de desbordamiento. efecto de la intervención en función del número de meses transcurridos desde el discurso. Las siguientes cuatro líneas simplemente dibujan un gráfico de los valores de nuestra serie original y administran algunas de las características del gráfico. Por último, agregamos una línea discontinua que muestra el efecto de la intervención manteniendo constante todo lo demás. El resultado se muestra en la Fig.  9.3. Como muestra la figura, el resultado de esta intervención es un salto grande y positivo en el número esperado de noticias, que se prolonga durante unos meses, pero finalmente regresa a los niveles previos a la intervención. Este tipo de gráfico es esencial para comprender cómo la intervención dinámica afecta realmente al modelo. Abrir imagen en nueva ventanaFigura 9.3 Figura 9.3 La línea discontinua muestra el efecto dinámico del discurso de Nixon de noviembre de 1973, manteniendo todo lo demás igual. La línea continua muestra los valores observados de la serie.

Como gráfico final para complementar nuestra visión del efecto de la intervención dinámica, podríamos dibujar un gráfico que muestre qué tan bien se alinean las predicciones del modelo completo con los valores reales de la serie. Podríamos hacer esto con el siguiente código:

meses <-c (1: 180)

full.pred <-pres.energy $ Energy-dynamic.mod $ residuales

plot (y = full.pred, x = months, xlab = “Month,”

ylab = "Historias de políticas energéticas", type = "l",  
  
 ylim = c (0,225), ejes = F)

puntos (y = energía pres. $ Energía, x = meses, pch = 20)

leyenda (x = 0, y = 200, leyenda = c (“Previsto,” “Verdadero”),

pch = c (NA, 20), lty = c (1, NA))

eje (1, en = c (1,37,73,109,145), etiquetas = c (“Enero de 1969,”

"Ene. 1972", "Ene. 1975", "Ene. 1978", "Ene. 1981"))

eje (2)

caja()

Nuevamente, comenzamos creando un índice de tiempo, meses . En la segunda línea, creamos nuestros valores predichos restando los residuos de los valores verdaderos. En la tercera línea de código, dibujamos un gráfico lineal de los valores predichos del modelo. En la cuarta línea, agregamos puntos que muestran los valores reales de la serie observada. Las líneas restantes completan el formato del gráfico. El gráfico resultante se muestra en la figura  9.4 . Como puede verse, el ajuste en la muestra es bueno, con los valores predichos siguiendo de cerca los valores verdaderos. Abrir imagen en nueva ventanaFigura 9.4 Figura 9.4 Valores predichos de un modelo de función de transferencia completo en una línea, con valores observados reales como puntos

Como punto final aquí, se anima al lector a consultar el código en la Sección. 9.5 para una sintaxis alternativa para producir las Figs. 9.3 y 9.4 . La compensación de la forma alternativa de dibujar estas figuras es que requiere más líneas de código por un lado, pero por otro lado, es más generalizable y más fácil de aplicar a su propia investigación. Además, el código alternativo presenta cómo el comando ts permite a los analistas convertir una variable en un objeto de serie temporal . Al ver a ambos enfoques es que vale la pena para ilustrar que, en general, muchas tareas se pueden realizar de muchas maneras en R .

9.2 Extensiones a modelos de regresión lineal por mínimos cuadrados Un segundo enfoque para el análisis de series de tiempo se basa más en la literatura econométrica y busca formas de extender los modelos de regresión lineal para tener en cuenta los problemas únicos asociados con los datos referenciados en el tiempo. Dado que ya discutimos ampliamente la visualización con estos datos en la Sect. 9.1 , no revisaremos los problemas de gráficos aquí. Sin embargo, al igual que con los modelos de tipo Box-Jenkins, el analista siempre debe comenzar dibujando una gráfica lineal de la serie de interés e idealmente también algunos predictores clave. Incluso los gráficos de diagnóstico como el ACF y PACF serían apropiados, además de los diagnósticos residuales como los que discutiremos en breve.

Al modelar datos desde un enfoque econométrico, los investigadores nuevamente tienen que decidir si usar una especificación estática o dinámica del modelo. Para los modelos estáticos en los que los valores actuales de los predictores afectan los valores actuales del resultado, los investigadores pueden estimar el modelo con mínimos cuadrados ordinarios (MCO) en el raro caso de que no haya correlación serial. Sin embargo, para las ganancias de eficiencia en un modelo estático, los mínimos cuadrados generalizados factibles (FGLS) son un mejor estimador. Por el contrario, en el caso de una forma funcional dinámica, se puede introducir una estructura de retardo en la especificación del modelo lineal.

Comenzando con modelos estáticos, el tipo de modelo más simple (aunque raramente apropiado) sería estimar el modelo usando MCO simple. Volviendo a los datos de nuestra política energética, la especificación de nuestro modelo aquí sería:

static.ols <-lm (Energía ~ rmn1173 + grf0175 + grf575 + jec477 +

jec1177 + jec479 + embargo + rehenes + oilc +  
  
 Aprobación + Desempleo, datos = energía pres.)

Al escribir resumen (static.ols) obtenemos nuestro resultado familiar de un modelo de regresión lineal. Sin embargo, tenga en cuenta que si existe una correlación en serie en las perturbaciones, estas estimaciones de los errores estándar son incorrectas:

Llamada:

lm (fórmula = Energía ~ rmn1173 + grf0175 + grf575 + jec477 +

jec1177 +

jec479 + embargo + rehenes + oilc + Aprobación +  
  
 Desempleo  
  
datos = pres.energy)

Derechos residuales de autor:

Mín. 1T Mediana 3T Máx.

-104,995 -12,921 -3,448 8,973 111,744

Coeficientes:

Estimar Std. Valor t de error Pr (> | t |)

(Intercepción) 319.7442 46.8358 6.827 1.51e-10 \*\*\*

rmn1173 78,8261 28,8012 2,737 0,00687 \*\*

grf0175 60.7905 26.7006 2.277 0.02406 \*

grf575 -4,2676 26,5315 -0,161 0,87240

jec477 47.0388 26.6760 1.763 0.07966.

jec1177 15.4427 26.3786 0.585 0.55905

jec479 72.0519 26.5027 2.719 0.00724 \*\*

embargo 96.3760 13.3105 7.241 1.53e-11 \*\*\*

rehenes -4,5289 7,3945 -0,612 0,54106

aceitec -5.8765 1.0848 -5.417 2.07e-07 \*\*\*

Aprobación -1.0693 0.2147 -4.980 1.57e-06 \*\*\*

Desempleo -3.7018 1.3861 -2.671 0.00831 \*\*

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05. 0,1 1

Error estándar residual: 26,26 en 168 grados de

libertad

R cuadrado múltiple: 0,5923, R cuadrado ajustado:

0.5656

Estadístico F: 22,19 en 11 y 168 DF, valor p:

<2.2e-16

Sin embargo, necesitamos diagnosticar si la correlación serial está presente en los residuos antes de estar contentos con los resultados. Para hacer esto, necesitamos cargar el paquete lmtest (presentado por primera vez en el Capítulo  6 ) para que algunos diagnósticos estén disponibles. Al cargar este paquete, podemos calcular una prueba de Durbin-Watson o Breusch-Godfrey para la autocorrelación:

biblioteca (lmtest)

dwtest (static.ols)

bgtest (static.ols)

Para los comandos dwtest y bgtest , simplemente proporcionamos el nombre del modelo como argumento principal. El D urbin- W atson prueba (calculado con dwtest ) prueba de correlación serial de primer orden y no es válido para un modelo que incluye una variable dependiente rezagada. Nuestra prueba produce el siguiente resultado:

Prueba de Durbin-Watson

datos: static.ols

DW = 1,1649, valor p = 1,313e-09

hipótesis alternativa: la verdadera autocorrelación es mayor

de 0

El estadístico d de Durbin-Watson (1.1649 en este caso), no tiene una distribución paramétrica. Tradicionalmente, el valor de d se ha comparado con tablas basadas en los resultados de Monte Carlo para determinar la importancia. Sin embargo, R proporciona un valor p aproximado con el estadístico. Para una prueba de Durbin-Watson, la hipótesis nula es que no hay autocorrelación, por lo que nuestro valor significativo de d sugiere que la autocorrelación es un problema.

Mientras tanto, los resultados de nuestro B reusch- G odfrey prueba (calculadas con bgtest ) ofrecer una conclusión similar. La prueba de Breusch-Godfrey tiene una distribución χ 2 y se puede utilizar para probar la autocorrelación en un modelo con una variable dependiente rezagada. De forma predeterminada, el comando bgtest comprueba la correlación en serie de primer orden, aunque la correlación en serie de orden superior se puede probar con la opción de orden . Nuestro resultado en este caso es:

Prueba de Breusch-Godfrey para la correlación serial de

ordenar hasta

1

datos: static.ols

Prueba LM = 38.6394, gl = 1, valor p = 5.098e-10

Nuevamente, la hipótesis nula es que no hay autocorrelación, por lo que nuestro valor significativo de χ 2 muestra que la correlación serial es una preocupación, y debemos hacer algo para explicar esto.

En este punto, podemos sacar una de dos conclusiones: la primera posibilidad es que la especificación de nuestro modelo estático sea ​​correcta, y necesitamos encontrar un estimador que sea eficiente en presencia de autocorrelación de errores. La segunda posibilidad es que hemos pasado por alto un efecto dinámico y necesitamos volver a especificar nuestro modelo. (En otras palabras, si hay un verdadero efecto de desbordamiento y no lo hemos modelado, los errores parecerán estar correlacionados en serie). Consideraremos cada posibilidad.

Primero, si estamos seguros de que nuestra especificación estática es correcta, entonces nuestra forma funcional es correcta, pero según el teorema de Gauss-Markov, OLS es ineficiente con autocorrelación de errores y los errores estándar están sesgados. Como alternativa, podemos utilizar mínimos cuadrados generalizados factibles (FGLS), que estima el nivel de correlación del error y lo incorpora al estimador. Aquí hay una variedad de técnicas de estimación, incluidos los estimadores de Prais-Winsten y Cochrane-Orcutt. Procedemos ilustrando el estimador Cochrane-Orcutt, aunque los usuarios deben tener cuidado con los supuestos del modelo. 9 En resumen, Cochrane - Orcuttvuelve a estimar el modelo varias veces, actualizando la estimación de la autocorrelación del error cada vez, hasta que converge a una estimación estable de la correlación. Para implementar este procedimiento en R , necesitamos instalar y luego cargar el paquete orcutt :

install.packages (“orcutt”)

biblioteca (orcutt)

cochrane.orcutt (static.ols)

Una vez que hemos cargado este paquete, insertamos el nombre de un modelo lineal que hemos estimado con OLS en el comando cochrane.orcutt . Esto luego vuelve a estimar iterativamente el modelo y produce nuestros resultados FGLS de la siguiente manera:

$ Cochrane.Orcutt

Llamada:

lm (fórmula = YB ~ XB - 1)

Derechos residuales de autor:

Mín. 1T Mediana 3T Máx.

-58.404 -9.352 -3.658 8.451 100.524

Coeficientes:

Estimar Std. Valor t de error Pr (> | t |)

XB (intercepción) 16.8306 55.2297 0.305 0.7609

XBrmn1173 91,3691 15,6119 5,853 2,5e-08 \*\*\*

XBgrf0175 32.2003 16.0153 2.011 0.0460 \*

XBgrf575 -7,9916 15,7288 -0,508 0,6121

XBjec477 29.6881 15.6159 1.901 0.0590.

XBjec1177 -6,4608 15,6174 -0,414 0,6796

XBjec479 -20.0677 15.6705 -1.281 0.2021

XBembargo 34.5797 15.0877 2.292 0.0232 \*

XBhostages -16.9183 14.1135 -1.199 0.2323

XBoilc 0,8240 1,0328 0,798 0,4261

Aprobación XBA -0,2399 0,2742 -0,875 0,3829

XB Desempleo -0,1332 4,3786 -0,030 0,9758

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05. 0,1 1

Error estándar residual: 20,19 en 167 grados de

libertad

R cuadrado múltiple: 0,2966, R cuadrado ajustado:

0.2461

Estadístico F: 5,87 en 12 y 167 DF, valor de p:

1.858e-08

$ rho

[1] 0,8247688

$ number.interaction

[1] 15

La primera parte de la tabla se parece a la salida de regresión lineal familiar (aunque las letras XB aparecen antes del nombre de cada predictor). Todos estos coeficientes, errores estándar y estadísticas inferenciales tienen exactamente la misma interpretación que en un modelo estimado con MCO, pero nuestras estimaciones ahora deberían ser eficientes porque se calcularon con FGLS. Cerca de la parte inferior de la salida, vemos $ rho , que nos muestra nuestra estimación final de autocorrelación de errores. También vemos $ number.interaction , que nos informa que el modelo se volvió a estimar en 15 iteraciones antes de que convergiera al resultado final. FGLS está destinado a producir estimaciones eficientes si una especificación estática es correcta.

Por el contrario, si creemos que una especificación dinámica es correcta, debemos trabajar para volver a especificar nuestro modelo lineal para capturar esa forma funcional. De hecho, si nos equivocamos en la forma funcional, nuestros resultados están sesgados, por lo que hacerlo bien es fundamental. Agregar una especificación de retraso a nuestro modelo puede ser considerablemente más fácil si instalamos y cargamos el paquete dyn . Nombramos nuestro modelo koyck.ols por razones que serán evidentes en breve:

install.packages (“dyn”)

biblioteca (dyn)

pres.energy <-ts (pres.energy)

koyck.ols <-dyn $ lm (Energía ~ retraso (Energía, -1) + rmn1173 +

grf0175 + grf575 + jec477 + jec1177 + jec479 + embargo +  
  
 rehenes + petróleoc + Aprobación + Desempleo, datos = energía pres.)

Después de cargar dyn , la segunda línea usa el comando ts para declarar que nuestros datos son datos de series de tiempo. En la tercera línea, observe que cambiamos el comando del modelo lineal para que lea, dyn $ lm . Esta modificación nos permite incluir variables rezagadas dentro de nuestro modelo. En particular, ahora hemos agregado rezago (Energía, -1) , que es el valor rezagado de nuestra variable dependiente. Con el comando lag , especificamos la variable que se retrasa y cuántas veces se retrasa. Al especificar -1 , buscamos el valor inmediatamente anterior. (Los valores positivos representan valores futuros). El retraso predeterminado es 0, que solo devuelve valores actuales.

Podemos ver los resultados de este modelo escribiendo resumen (koyck.ols) :

Llamada:

lm (fórmula = dyn (Energía ~ retraso (Energía, -1) + rmn1173 +

grf0175 +

grf575 + jec477 + jec1177 + jec479 + embargo +

rehenes +  
  
petróleoc + Aprobación + Desempleo), datos = energía pres.)

Derechos residuales de autor:

Mín. 1T Mediana 3T Máx.

-51.282 -8.638 -1.825 7.085 70.472

Coeficientes:

Estimar Std. Valor t de error Pr (> | t |)

(Intercepción) 62.11485 36.96818 1.680 0.09479.

rezago (Energía, -1) 0,73923 0,05113 14,458 <2e-16 \*\*\*

rmn1173 171.62701 20.14847 8.518 9.39e-15 \*\*\*

grf0175 51.70224 17.72677 2.917 0.00403 \*\*

grf575 7.05534 17.61928 0.400 0.68935

jec477 39.01949 17.70976 2.203 0.02895 \*

jec1177 -10.78300 17.59184 -0.613 0.54075

jec479 28.68463 17.83063 1.609 0.10958

embargo 10.54061 10.61288 0.993 0.32206

rehenes -2.51412 4.91156 -0.512 0.60942

aceitec -1.14171 0.81415 -1.402 0.16268

Aprobación -0,15438 0,15566 -0,992 0,32278

Desempleo -0,88655 0,96781 -0,916 0,36098

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05. 0,1 1

Error estándar residual: 17,42 en 166 grados de

libertad

(2 observaciones eliminadas por falta de información)

R cuadrado múltiple: 0,822, R cuadrado ajustado:

0,8092

Estadístico F: 63,89 en 12 y 166 DF, valor de p:

<2.2e-16

Nuestra especificación aquí a menudo se denomina modelo Koyck. Esto se debe a que Koyck (1954) observó que cuando se incluye una variable dependiente rezagada como predictor, cada predictor tendrá efectos de desbordamiento en los meses siguientes.

Considere dos ejemplos de efectos de desbordamiento de predictores. Primero, nuestro coeficiente para el habla de Nixon es aproximadamente 172. Aquí, estamos interesados ​​en un efecto de impulso por el cual el predictor aumentó a 1 en el mes en que se dio el habla, y luego volvió a 0. Por lo tanto, en el mes de noviembre de 1973 cuando se pronunció el discurso, el efecto esperado de este discurso manteniendo todo lo demás igual es un aumento de 172 historias en la cobertura de la política energética. Sin embargo, en diciembre de 1973, el nivel de cobertura de noviembre es un predictor, y la cobertura de noviembre fue moldeada por el discurso. Dado que nuestro coeficiente en la variable dependiente rezagada es aproximadamente 0,74, y desde 0. 74 × 172 ≈ 128, por lo tanto, esperamos que el habla aumente la cobertura de energía en diciembre en 128, ceteris paribus. Sin embargo, el efecto también persistiría en enero porque el valor de diciembre predice el valor de enero de 1974. Desde 0. 74 × 0. 74 × 172 ≈ 94, esperamos que el efecto del discurso de Nixon sea un aumento de 94 pisos en la cobertura de energía en enero, ceteris paribus . Este tipo de efecto dinámico decadente persiste para un efecto de impulso en cualquiera de estas variables, por lo que esta es una forma poderosa de especificar un modelo dinámico. Además, tenga en cuenta que un investigador podría trazar fácilmente estos efectos de la intervención en descomposición a lo largo del tiempo para visualizar el impacto dinámico de una intervención. Este gráfico es válido para un modelo de Koyck y se parecería al resultado de la figura  9.3 .

En segundo lugar, considere el efecto del desempleo. No deberíamos darle mucha importancia al impacto de este predictor porque el coeficiente no es discernible, pero sirve como ejemplo de interpretación del efecto dinámico de un predictor continuo. Si estamos interesados ​​en un efecto escalón , nos gustaría saber cuál sería el impacto a largo plazo si un predictor aumentara en una sola unidad y se mantuviera en ese nivel más alto. Entonces, aunque no se puede discernir estadísticamente, el coeficiente de desempleo es - 0. 89, lo que significa que un aumento de un punto porcentual en el desempleo disminuye la atención de las noticias a la política energética en casi nueve décimas partes de una historia, en el mismo mes, en promedio, y ceteris paribus . Pero si el desempleo se mantuvo un punto porcentual más alto, ¿cómo cambiaría la cobertura a largo plazo? Si β13 es el coeficiente de desempleo y β 2 es el coeficiente de la variable dependiente rezagada, entonces el efecto a largo plazo se calcula mediante (Keele y Kelly 2006, pag. 189): β131 -β2 (9,2) Podemos calcular esto en R simplemente haciendo referencia a nuestros coeficientes: koyck.ols $ coeficientes [13] / (1-koyck.ols $ coeficientes [2])

Nuestra producción es - 3. 399746, lo que significa que un aumento persistente de 1% en el desempleo reduciría la cobertura de noticias de televisión sobre política energética a largo plazo en 3.4 historias en promedio y todo lo demás igual. Nuevamente, este tipo de efecto a largo plazo podría ocurrir para cualquier variable que no se limite a una entrada de pulso.

Como estrategia final, podríamos incluir uno o más rezagos de uno o más predictores sin incluir una variable dependiente rezagada. En este caso, cualquier derrame se limitará a lo que incorporemos directamente en la especificación del modelo. Por ejemplo, si solo quisiéramos un efecto dinámico del habla de Nixon y una especificación estática para todo lo demás, podríamos especificar este modelo:

udl.mod <-dyn $ lm (Energía ~ rmn1173 + retardo (rmn1173, -1) +

retraso (rmn1173, -2) + retraso (rmn1173, -3) + retraso (rmn1173, -4) +  
  
 grf0175 + grf575 + jec477 + jec1177 + jec479 + embargo +  
  
 rehenes + petróleoc + Aprobación + Desempleo, datos = energía pres.)

En esta situación, hemos incluido el valor actual del indicador de habla de Nixon, así como cuatro rezagos. Para una intervención, eso significa que este predictor tendrá efecto en noviembre de 1973 y durante 4 meses después. (En abril de 1974, sin embargo, el efecto cae abruptamente a 0, donde permanece). Vemos los resultados de este modelo escribiendo resumen (udl.mod) :

Llamada:

lm (fórmula = dyn (Energía ~ rmn1173 + retraso (rmn1173, -1) + retraso

(rmn1173, -2) + retraso (rmn1173, -3) + retraso (rmn1173, -4)  
  
+ grf0175 + grf575 + jec477 + jec1177 + jec479 + embargo  
  
+ rehenes + petróleoc + Aprobación + Desempleo), datos = energía pres.)

Derechos residuales de autor:

Mín. 1T Mediana 3T Máx.

-43,654 -13,236 -2,931 7,033 111,035

Coeficientes:

Estimar Std. Valor t de error Pr (> | t |)

(Intercepción) 334.9988 44.2887 7.564 2.89e-12 \*\*\*

rmn1173 184.3602 34.1463 5.399 2.38e-07 \*\*\*

retraso (rmn1173, -1) 181.1571 34.1308 5.308 3.65e-07 \*\*\*

retraso (rmn1173, -2) 154.0519 34.2151 4.502 1.29e-05 \*\*\*

retraso (rmn1173, -3) 115.6949 34.1447 3.388 0.000885 \*\*\*

retraso (rmn1173, -4) 75.1312 34.1391 2.201 0.029187 \*

grf0175 60.5376 24.5440 2.466 0.014699 \*

grf575 -3.4512 24.3845 -0.142 0.887629

jec477 45.5446 24.5256 1.857 0.065146.

jec1177 14.5728 24.2440 0.601 0.548633

jec479 71.0933 24.3605 2.918 0.004026 \*\*

embargo -9.7692 24.7696 -0.394 0.693808

rehenes -4,8323 6,8007 -0,711 0,478392

aceitec -6.1930 1.0232 -6.053 9.78e-09 \*\*\*

Aprobación -1.0341 0.1983 -5.216 5.58e-07 \*\*\*

Desempleo -4,4445 1,3326 -3,335 0,001060 \*\*

Signif. códigos: 0 \*\*\* 0,001 \*\* 0,01 \* 0,05. 0,1 1

Error estándar residual: 24,13 en 160 grados de libertad

(8 observaciones eliminadas por falta de información)

R-cuadrado múltiple: 0,6683, R-cuadrado ajustado: 0,6372

Estadístico F: 21,49 en 15 y 160 DF, valor de p: <2,2e-16

Como era de esperar, el efecto se reduce cada mes después del inicio. Para este modelo de rezago distribuido sin restricciones, tenemos que estimar varios parámetros más de los que requiere el modelo de Koyck, pero en algunos casos puede tener sentido teóricamente.

9.3 Autorregresión vectorial El enfoque final que describiremos en este capítulo es la autorregresión vectorial (VAR). El enfoque VAR es útil cuando se estudian varias variables que son endógenas entre sí porque existe una causalidad recíproca entre ellas. El marco básico es estimar un modelo de regresión lineal para cada una de las variables endógenas. En cada modelo lineal, incluir varios valores retardados de la variable de resultado en sí (digamos p rezagos de la variable), así como p rezagos de todas las otras variables endógenas. Así que para el caso simple de dos variables endógenas, X e Y , en los que nos planteamos nuestra longitud del rezago que p  = 3, queremos estimar dos ecuaciones que pueden ser representados de la siguiente manera: ytXt==β0+β1yt - 1+β2yt - 2+β3yt - 3+β4Xt - 1+β5Xt - 2+β6Xt - 3+ϵtγ0+γ1Xt - 1+γ2Xt - 2+γ3Xt - 3+γ4yt - 1+γ5yt - 2+γ6yt - 3+δt (9,3) Un tratamiento más completo de esta metodología, incluida la notación para modelos con variables más endógenas y ejemplos de modelos que también incluyen variables exógenas, se puede encontrar en Brandt y Williams (2007). Con este modelo, se pueden utilizar pruebas como las de causalidad de Granger para evaluar si existe un efecto causal de una variable endógena a otra (Granger 1969). Como ejemplo de cómo implementar un modelo VAR en R , pasamos al trabajo de Brandt y Freeman (2006), quienes analizan datos semanales sobre el conflicto israelo-palestino. Sus datos se extraen del Kansas Event Data System, que codifica automáticamente los informes de noticias en inglés para medir los eventos políticos, con el objetivo de utilizar esta información como una advertencia temprana para predecir cambios políticos. Las variables endógenas en estos datos son acciones políticas escaladas tomadas por EE. UU., Israel o Palestina, y dirigidas a uno de los otros actores. Esto produce seis variables a2i , a2p , i2a , p2a , i2p y p2i . Las abreviaturas son “a” para estadounidense, “i” para israelí y “p” para palestino. Como ejemplo, i2pmide el valor escalado de las acciones israelíes dirigidas a los palestinos. Los datos semanales que usaremos van desde el 15 de abril de 1979 al 26 de octubre de 2003, por un total de 1278 semanas.

Para continuar, necesitamos instalar y cargar el paquete vars para que los comandos de estimación relevantes estén disponibles. También necesitamos el paquete externo porque nuestros datos están en formato Stata: 10

install.packages (“vars”)

biblioteca (vars)

biblioteca (extranjera)

levant.0 <- read.dta (“levant.dta”)

levant <- subconjunto (levant.0,

select = c ("a2i", "a2p", "i2p", "i2a", "p2i", "p2a"))

Después de cargar los paquetes, cargamos nuestros datos en la tercera línea, nombrándola levant.0 . Estos datos también contienen tres índices relacionados con la fecha, por lo que, para fines de análisis, en realidad necesitamos crear una segunda copia de los datos que solo incluya nuestras seis variables endógenas sin los índices. Hacemos esto usando el comando subconjunto para crear el conjunto de datos llamado levant .

Un paso clave en este punto es elegir la longitud de retraso adecuada, p . La longitud del retraso debe capturar todos los procesos de error y la dinámica causal. Un enfoque para determinar la longitud de retardo apropiada es ajustar varios modelos, elegir el modelo con el mejor ajuste y luego ver si los residuos muestran alguna evidencia de correlación serial. El comando VARselect calcula automáticamente varias v ector un uto r modelos egression y seleccione s que se retrasan longitud tiene el mejor ajuste. Dado que nuestros datos son semanales, consideramos retrasos de hasta 104 semanas, o datos de 2 años, para considerar todas las opciones posibles. El siguiente comando puede tardar un minuto en ejecutarse porque se están estimando 104 modelos:

levant.select <-VARselect (levant, type = “const,” lag.max = 104)

Para saber cuál de las longitudes de retardo se ajusta mejor, podemos escribir levant.select $ selection . Nuestro resultado es simplemente el siguiente:

AIC (n) HQ (n) SC (n) FPE (n)

60 4 1 47

Esto informa la longitud de retraso elegida para el criterio de información de Akaike, el criterio de información de Hannan-Quinn, el criterio de Schwarz y el error de predicción de pronóstico. Los cuatro índices están codificados para que los valores más bajos sean mejores. Para contrastar los extremos, el valor más bajo del criterio de Schwarz, que tiene una fuerte penalización por parámetros adicionales, es para el modelo con solo un rezago de las variables endógenas. Por el contrario, el mejor ajuste en el AIC proviene del modelo que requiere 60 rezagos, lo que quizás indique que existe una estacionalidad anual. Se puede ver una impresión mucho más larga con el valor de cada uno de los cuatro índices de ajuste para los 104 modelos escribiendo: levant.select $ criteria. Idealmente, nuestros índices de ajuste se habrían establecido en modelos con longitudes de retardo similares. Como no lo hicieron, y dado que tenemos 1278 observaciones, tomaremos la ruta más segura con el largo retraso sugerido por la AIC. 11

Para estimar la v ector un uto r modelo egression con p  = 60 retardos de cada variable endógena, escribimos:

levant.AIC <-VAR (levant, tipo = “const,” p = 60)

El comando VAR requiere el nombre de un conjunto de datos que contiene todas las variables endógenas. Con la opción de tipo , hemos elegido “const” en este caso (el predeterminado). Esto significa que cada uno de nuestros modelos lineales incluye una constante. (Otras opciones incluyen especificar una “tendencia,” “tanto” una constante como una tendencia, o “ninguna” que no incluye ninguna.) Con la opción p elegimos la longitud del rezago. Una opción que no usamos aquí es la opción exógena , que nos permite especificar variables exógenas.

Una vez que hemos estimado nuestro modelo usando VAR , lo siguiente que debemos hacer es diagnosticar el modelo usando una llamada especial a la función plot :

parcela (levant.AIC, lag.acf = 104, lag.pacf = 104)

Al utilizar el nombre de nuestro modelo VAR, levant.AIC , como único argumento en la gráfica , R proporcionará automáticamente una gráfica de diagnóstico para cada una de las variables endógenas. Con las opciones de lag.acf y lag.pacf , especificamos que las gráficas de ACF y PACF que los informes de este comando deben mostrarnos el valor de 2 años (104 semanas) de patrones de autocorrelación. Para nuestros datos, R produce seis gráficos. R muestra estos gráficos uno a la vez, y entre cada uno, la consola presentará el siguiente mensaje:

Presione para ver la siguiente trama:

R mantendrá un gráfico en el visor hasta que presione la tecla Retorno , momento en el que pasa al siguiente gráfico. Este proceso se repite hasta que se muestra el gráfico de cada resultado.

Alternativamente, si quisiéramos ver la gráfica de diagnóstico para una variable endógena en particular, podríamos escribir:

plot (levant.AIC, lag.acf = 104, lag.pacf = 104, names = “i2p”)

Aquí, la opción de nombres nos ha permitido especificar que queremos ver los diagnósticos de i2p (acciones israelíes dirigidas hacia Palestina). La gráfica de diagnóstico resultante se muestra en la figura 9.5 . . El gráfico tiene cuatro partes: En la parte superior hay un gráfico de líneas que muestra los valores reales en una línea negra sólida y los valores ajustados en una línea discontinua azul. Directamente debajo de esto hay un gráfico lineal de los residuos contra una línea en cero. Estos dos primeros gráficos pueden ilustrar si el modelo hace predicciones insesgadas del resultado de manera consistente y si los residuos son homocedásticos a lo largo del tiempo. En general, los pronósticos se sitúan constantemente alrededor de cero, aunque en las últimas 200 observaciones la varianza del error parece aumentar ligeramente. El tercer gráfico, en la parte inferior izquierda, es el ACF para los residuos en i2p . 12 Por último, en la parte inferior derecha del panel, vemos el PACF para i2pResiduos de. Ningún pico es significativo en el ACF y solo un pico es significativo en el PACF durante 2 años, por lo que concluimos que nuestra estructura de rezagos ha filtrado suficientemente cualquier correlación serial en esta variable. Abrir imagen en nueva ventanaFigura 9.5 Figura 9.5 Valores predichos, función de autocorrelación residual y función de autocorrelación parcial residual para la serie Israel-Palestina en un modelo de autorregresión vectorial de seis variables

Al interpretar un modelo VAR, recurrimos a dos herramientas para extraer inferencias e interpretaciones de estos modelos. Primero, usamos la prueba de causalidad de Granger para determinar si una variable endógena causa las otras. Esta prueba es simplemente una prueba F de bloque para determinar si todos los rezagos de una variable pueden excluirse del modelo. Para una prueba conjunta de si una variable afecta a las otras variables en el sistema, podemos usar el comando de causalidad . Por ejemplo, si quisiéramos probar si las acciones israelíes hacia Palestina causaron acciones de las otras cinco díadas dirigidas, escribiríamos:

causalidad (levant.AIC, cause = “i2p”) $ Granger

El comando aquí solicita el nombre del modelo primero ( levant.AIC ), y luego con la opción de causa , especificamos cuál de las variables endógenas deseamos probar el efecto. Nuestro resultado es el siguiente:

Causalidad Granger H0: i2p no causa Granger

a2i a2p i2a

p2i p2a

datos: VAR objeto levant.AIC

Prueba F = 2.1669, df1 = 300, df2 = 5142, valor p

<2.2e-16

La hipótesis nula es que los coeficientes para todos los rezagos de i2p son cero cuando se modela cada una de las otras cinco variables de resultado. Sin embargo, nuestra prueba F es significativa aquí, como muestra el minúsculo valor p . Por lo tanto, concluiríamos que las acciones israelíes dirigidas hacia Palestina tienen un efecto causal sobre las otras variables de acción política en el sistema.

Podemos proceder a probar si cada una de las otras cinco variables de Granger causan los otros predictores en el sistema considerando cada una con el comando de causalidad :

causalidad (levant.AIC, cause = “a2i”) $ Granger

causalidad (levant.AIC, cause = “a2p”) $ Granger

causalidad (levant.AIC, cause = “i2a”) $ Granger

causalidad (levant.AIC, cause = “p2i”) $ Granger

causalidad (levant.AIC, cause = “p2a”) $ Granger

Los resultados no se reproducen aquí para preservar el espacio. Sin embargo, en el nivel de confianza del 95%, verá que cada variable causa significativamente a las demás, a excepción de las acciones estadounidenses dirigidas hacia Israel ( a2i ).

Finalmente, para tener una idea del impacto sustantivo de cada predictor, recurrimos al análisis de respuesta al impulso. La lógica aquí es algo similar al análisis de intervención que graficamos en la figura  9.3 . Una función de respuesta al impulso considera un aumento de una unidad en una de las variables endógenas y calcula cómo tal choque exógeno influiría dinámicamente en todas las variables endógenas, dados los patrones de autorregresión y dependencia.

Al interpretar una función de impulso-respuesta, una consideración clave es el hecho de que los choques a una variable endógena casi siempre están correlacionados con los choques a otras variables. Por lo tanto, debemos considerar que es probable que un choque de una unidad en el término residual de una variable cree un movimiento contemporáneo en los residuos de las otras variables. Los términos fuera de la diagonal de la matriz de varianza-covarianza de los residuos de las variables endógenas, Σ^ , nos dice cuánto covarían los choques.

Hay varias formas de abordar este problema. Uno es usar la teoría para determinar el orden de las variables endógenas. En este enfoque, el investigador asume que un choque en una variable endógena no se ve afectado por choques en ninguna otra variable. Entonces, los choques de una segunda variable se ven afectados solo por los de la primera variable y no por los demás. El investigador repite de forma recursiva este proceso para identificar un sistema en el que todas las variables se pueden ordenar en una cadena causal. Con un sistema diseñado teóricamente como este, el investigador puede determinar cómo los choques contemporáneos se afectan entre sí con una descomposición estructurada de Cholesky de Σ^ (Enders 2009, pag. 309). Una segunda opción, que es la opción predeterminada en el comando irf de R , es asumir que no hay conocimiento teórico del orden causal de los choques contemporáneos y aplicar el método de ortogonalización de los residuos . Esto implica otra descomposición de Cholesky, en la que encontramos A- 10 resolviendo A- 1 ′0A0=Σ^ . Para obtener más detalles sobre el orden de respuesta o la ortogonalización de residuos, consulte Brandt y Williams (2007, págs. 36-41 y 66-70), Enders (2009, págs. 307-315) o Hamilton (1994, págs. 318-324).

Como ejemplo de una función de respuesta al impulso, graficaremos el efecto de un evento político adicional de Israel dirigido hacia Palestina. R calculará nuestra i mpulse r espuesta f unción con el IRF del sistema, utilizando la ortogonalización por defecto de los residuos. Una de las opciones clave de este comando es el arranque , que determina si se debe construir un intervalo de confianza con correas de arranque . Generalmente, es aconsejable informar la incertidumbre en las predicciones, pero el proceso puede tardar varios minutos. 13 Entonces, si el lector desea un resultado rápido, configure boot = FALSE . Para obtener el resultado con intervalos de confianza, escriba:

levant.irf <-irf (levant.AIC, impulso = “i2p,” n.ahead = 12, boot = TRUE)

Llamamos a nuestra función de respuesta al impulso levant.irf . El comando requiere que digamos el nombre de nuestro modelo ( levant.AIC ). La opción impulso nos pide que nombremos la variable de la que queremos el efecto, la opción n. Anticipada establece con qué anticipación deseamos pronosticar (decimos 12 semanas o 3 meses), y finalmente boot determina si crear intervalos de confianza basados ​​en una muestra de bootstrap.

Una vez que hayamos calculado esto, podemos graficar la función de respuesta al impulso con una llamada especial para graficar :

parcela (levant.irf)

El gráfico resultante se muestra en la figura  9.6 . Hay seis paneles, uno para cada una de las seis variables endógenas. El eje vertical de cada panel enumera el nombre de la variable endógena y representa el cambio esperado en esa variable. El eje horizontal de cada panel representa el número de meses que han transcurrido desde el choque. Cada panel muestra una línea roja sólida en cero, que representa dónde cae un no efecto en el gráfico. La línea negra continua representa el efecto esperado en cada mes y las líneas discontinuas rojas representan cada intervalo de confianza. Como muestra la figura, el impacto real de una conmoción en las acciones de Israel a Palestina es dramático para la propia serie Israel a Palestina ( i2p) debido a la autorregresión y la retroalimentación de los efectos a otras series. También vemos un salto significativo en las acciones de Palestina a Israel ( p2i ) durante los siguientes 3 meses. Con las otras cuatro series, los efectos palidecen en comparación. Fácilmente podríamos producir gráficos similares a la figura  9.6 calculando la función de respuesta al impulso para un choque en cada una de las seis entradas. En general, esta es una buena idea, pero se omite aquí por espacio. Abrir imagen en nueva ventanaFigura 9.6 Figura 9.6 Función de respuesta al impulso para un choque de una unidad en la serie Israel-Palestina en un modelo de autorregresión vectorial de seis variables

9.4 Lecturas adicionales sobre el análisis de series de tiempo Con estas herramientas en la mano, el lector debe tener alguna idea de cómo estimar e interpretar modelos de series de tiempo en R usando tres enfoques: modelado de Box-Jenkins, modelado econométrico y autorregresión vectorial. Tenga en cuenta que, si bien este capítulo utiliza ejemplos simples, el análisis de series de tiempo generalmente es un desafío. Puede ser difícil encontrar un buen modelo que represente adecuadamente todos los procesos de tendencia y error, pero el analista debe trabajar con cuidado en todos estos problemas o las inferencias estarán sesgadas. (Vean de nuevo, Granger y Newbold 1974.) Por lo tanto, asegúrese de reconocer que a menudo se necesitan varios intentos para encontrar un modelo que tenga en cuenta todos los problemas.

También vale la pena tener en cuenta que este capítulo no puede abordar de manera integral ninguno de los tres enfoques que consideramos, y mucho menos abordar todos los tipos de análisis de series de tiempo. (El análisis espectral, el análisis de ondículas, los modelos de espacio de estados y los modelos de corrección de errores son solo algunos de los temas que no se tratan aquí). Por lo tanto, se recomienda al lector interesado que consulte otros recursos para obtener más detalles importantes sobre varios temas en el modelado de series de tiempo. Buenos libros que cubren una variedad de temas de series de tiempo e incluyen código R son: Cowpertwait y Metcalfe (2009), Cryer y Chan (2008) y Shumway y Stoffer (2006). Para obtener más profundidad sobre la teoría detrás de las series de tiempo, los buenos volúmenes de científicos políticos incluyen Box-Steffensmeier et al. (2014) y Brandt y Williams (2007). Otros buenos libros que cubren la teoría de series de tiempo incluyen: Box et al. 2008, Enders (2009), Wei (2006), Lütkepohl (2005), y para una versión avanzada, consulte Hamilton (1994). Varios libros se centran en temas específicos y en la aplicación de los métodos en R : por ejemplo, Petris et al. (2009) cubre modelos lineales dinámicos, también llamados modelos de espacio de estado, y explica cómo usar el paquete dlm correspondiente en R para aplicar estos métodos. Pfaff (2008) analiza los modelos de datos cointegrados, como los modelos de corrección de errores y los modelos de corrección de errores vectoriales, utilizando el paquete R vars . Además, los lectores interesados ​​en modelos de sección transversal de series de tiempo, o datos de panel, deben consultar el paquete plm , que facilita la estimación de modelos apropiados para esos métodos. Mientras tanto, Mátyás y Sevestre (2008) ofrece una base teórica sobre métodos de datos de panel.

9.5 Código de serie temporal alternativo Como se menciona en la Secta. 9.1 , ahora mostraremos una sintaxis alternativa para producir las Figs. 9.3 y 9.4 . Este código es un poco más largo, pero es más generalizable. 14 En primer lugar, si no tiene todos los paquetes, objetos de datos y modelos cargados de antes, asegúrese de volver a cargar algunos de ellos para que podamos dibujar la figura:

biblioteca (TSA)

pres.energy <-read.csv (“PESenergy.csv”)

predictores <-as.matrix (subconjunto (pres.energy, select = c (rmn1173,

grf0175, grf575, jec477, jec1177, jec479, embargo, rehenes,  
  
 oilc, Aprobación, Desempleo)))

Las tres líneas de código anteriores se ejecutaron anteriormente en el capítulo. A modo de recordatorio, la primera línea carga el paquete TSA , la segunda carga los datos de cobertura de nuestra política energética y la tercera crea una matriz de predictores.

Ahora, para volver a dibujar cualquiera de las figuras, debemos dedicarnos un poco a la gestión de datos:

meses <- 1: 180

static.predictors <- predictores [, - 1]

predictores.dinámicos <- predictores [, 1, drop = FALSE]

y <- ts (energía pres. $ Energía, frecuencia = 12, inicio = c (1972, 1))

Primero, definimos un índice mensual, como antes. En segundo lugar, subconjuntamos nuestros predictores matriciales con aquellos que tienen un efecto estático. En tercer lugar, aislamos nuestro predictor dinámico del discurso de Nixon. En cuarto lugar, utilizamos el comando ts para declarar la cobertura de la política energética como un objeto de serie temporal. En esta última línea, usamos la opción de frecuencia para especificar que estos son datos mensuales (de ahí el valor 12) y la opción de inicio para notar que estos datos comienzan en el primer mes de 1972.

A continuación, necesitamos estimar realmente nuestra función de transferencia. Una vez hecho esto, podemos guardar varias salidas del modelo:

dynamic.mod <-arimax (y, order = c (1,0,0), xreg = static.predictors,

xtransf = predictores dinámicos, transferencia = lista (c (1,0)))

b <- coef (mod dinámico)

static.coefs <- b [match (colnames (static.predictors), names (b))]

ma.coefs <- b [grep (“MA0,” nombres (b))]

ar.coefs <- b [grep (“AR1,” nombres (b))]

La primera línea reajusta nuestra función de transferencia. El segundo usa el comando coef para extraer los coeficientes del modelo y guardarlos en un vector llamado b . Las últimas tres líneas separan nuestros coeficientes en efectos estáticos ( static.coefs ), efectos dinámicos iniciales ( ma.coefs ) y términos de decaimiento ( ar.coefs ). En cada línea, hacemos referencia cuidadosamente a los nombres de nuestro vector de coeficientes, usando el comando match para encontrar coeficientes para los predictores estáticos, y luego el comando grep para buscar términos que contengan MA0 y AR1 , respectivamente, como términos de salida de una transferencia función hacer.

Con todos estos elementos extraídos, ahora pasamos específicamente a volver a dibujar la Fig.  9.3 , que muestra el efecto de la intervención del habla de Nixon contra los datos reales. Nuestro efecto de intervención consta de dos partes, el valor esperado de mantener todos los predictores estáticos en sus valores para el mes 58 y el efecto dinámico de la función de transferencia. Creamos esto de la siguiente manera:

xreg.pred <-b [“interceptar”] + static.coefs% \*% static.predictors [58,]

transf.pred <- as.numeric (predictores.dinámicos% \*% ma.coefs +

ma.coefs \* (ar.coefs ^ (meses-59)) \* (meses> 59))

y.pred <-ts (xreg.pred + transf.pred, frecuencia = 12, inicio = c (1972,1))

La primera línea simplemente hace la predicción estática a partir de una ecuación lineal. El segundo usa nuestros efectos iniciales y términos de decaimiento para predecir el efecto dinámico de la intervención. En tercer lugar, sumamos las dos piezas y las guardamos como una serie de tiempo con la misma frecuencia y fecha de inicio que la serie original. Con tanto y como y.pred ahora codificados como series de tiempo de la misma frecuencia durante el mismo lapso de tiempo, ahora es fácil recrear la Fig.  9.3 :

plot (y, xlab = “Month,” ylab = “Energy Policy Stories,” type = “l”)

líneas (y.pred, lty = 2, col = ‘blue,’ lwd = 2)

La primera línea simplemente traza la serie de tiempo original y la segunda línea agrega el efecto de intervención en sí.

Con todo el trabajo de configuración que hemos realizado, reproducir la Fig.  9.4 ahora solo requiere tres líneas de código:

full.pred <-ajustado (dynamic.mod)

plot (full.pred, ylab = “Historias de política energética,” type = “l,”

ylim = c (0,225))

puntos (y, pch = 20)

La primera línea simplemente usa el comando ajustado para extraer valores ajustados del modelo de función de transferencia. La segunda línea traza estos valores ajustados y la tercera agrega los puntos que representan la serie original.

9.6 Problemas de práctica Este conjunto de problemas de práctica revisa cada uno de los tres enfoques del modelado de series de tiempo presentados en el capítulo y luego plantea una pregunta adicional sobre los datos de energía de Peake y Eshbaugh-Soha que le pide que aprenda sobre un nuevo método. Las preguntas 1 a 3 se relacionan con modelos de una sola ecuación, por lo que todas estas preguntas utilizan un conjunto de datos sobre el consumo de electricidad en Japón. Mientras tanto, la pregunta n. ° 4 utiliza datos económicos de EE. UU. Para un modelo de ecuaciones múltiples. 1. Visualización de series de tiempo: Wakiyama et al. (2014) estudian el consumo de electricidad en Japón, evaluando si el accidente nuclear de Fukushima del 11 de marzo de 2011 afectó el consumo de electricidad en varios sectores. Lo hacen mediante la realización de un análisis de intervención sobre medidas mensuales de consumo eléctrico en megavatios (MW), de enero de 2008 a diciembre de 2012. Cargan el paquete externo y abren estos datos en formato Stata desde el archivo comprehensivoJapanEnergy.dta . Este archivo de datos está disponible en Dataverse (consulte la página vii) o en el contenido en línea de este capítulo (consulte la página 155). Nos centraremos en el consumo de electricidad de los hogares (nombre de la variable: casa). Tome el logaritmo de esta variable y trace una gráfica lineal del consumo de electricidad del hogar registrado de mes a mes. ¿Qué patrones son evidentes en estos datos?

1. Modelado de Box – Jenkins: un. Trace la función de autocorrelación y la función de autocorrelación parcial para el consumo de electricidad doméstico registrado en Japón. ¿Cuáles son las características más evidentes de estas figuras?
2. Wakiyama y col. (2014) argumentan que un ARIMA (1,0,1), con un componente ARIMA de temporada (1,0,0) se ajusta a esta serie. Estime este modelo e informe sus resultados. ( Sugerencia: para este modelo, querrá incluir la opción estacional = lista (orden = c (1,0,0), período = 12) en el comando arima ).
3. ¿Qué tan bien encaja este modelo ARIMA? ¿Cómo se ven el ACF y el PACF para los residuos de este modelo? ¿Cuál es el resultado de una prueba Q de Ljung-Box ?
4. Utilice el comando arimax del paquete TSA . Estime un modelo que utiliza el proceso de error ARIMA de antes, los predictores estáticos de temperatura ( temp ) y temperatura al cuadrado ( temp2 ), y una función de transferencia para la intervención de Fukushima ( ficticia ).
5. Bono: el indicador de Fukushima es en realidad una intervención escalonada , en lugar de un pulso . Esto significa que el efecto se acumula en lugar de decaer . La nota a pie de página 8 describe cómo se acumulan estos efectos. Haga un dibujo del efecto acumulativo de la intervención de Fukushima en el consumo de electricidad de los hogares registrado.
6. Modelado econométrico: un. Ajuste un modelo lineal estático utilizando OLS para el consumo de electricidad doméstico registrado en Japón. Utilice la temperatura ( temp ), la temperatura al cuadrado ( temp2 ) y el indicador de Fukushima ( ficticio ) como predictores. Cargue el paquete lmtest y calcule tanto una prueba de Durbin-Watson como una de Breusch-Godfrey para este modelo lineal. ¿Qué conclusiones sacarías de cada uno? ¿Por qué crees que obtienes este resultado?
7. Vuelva a estimar el modelo lineal estático de consumo eléctrico doméstico registrado utilizando FGLS con el algoritmo Cochrane-Orcutt. ¿Qué tan similares o diferentes son sus resultados de los resultados de OLS? ¿Por qué crees que es esto?
8. Cargue el paquete dyn y agregue una variable dependiente retrasada a este modelo. ¿Cuál de los tres modelos econométricos cree que es el más apropiado y por qué? ¿Cree que su modelo econométrico preferido o el análisis de intervención de Box-Jenkins es más apropiado? ¿Por qué?
9. Autorregresión vectorial: un. Enders (2009, pag. 315) presenta datos trimestrales sobre la economía de Estados Unidos, que va desde el segundo trimestre de 1959 hasta el primer trimestre de 2001. Cargue los paquetes vars y extranjeros , y luego abra los datos en formato Stata desde el archivo moneyDem.dta . Este archivo está disponible en Dataverse (consulte la página vii) o en el contenido en línea de este capítulo (consulte la página 155). Haga un subconjunto de los datos para incluir solo tres variables: cambio en el PIB real registrado ( dlrgdp ), cambio en la oferta monetaria real M2 ( dlrm2 ) y cambio en la tasa de interés a 3 meses de las letras del Tesoro de EE . UU . ( Drs ). Con el comando VARselect , determine la longitud de retardo que mejor se ajuste para un modelo VAR de estas tres variables, de acuerdo con el AIC.
10. Calcule el modelo que determinó que es el que mejor se ajusta según el AIC. Examine las gráficas de diagnóstico. ¿Cree que estas series están libres de correlación serial y que la forma funcional es correcta?
11. Para cada una de las tres variables, pruebe si la variable Granger causa las otras dos.
12. En política monetaria, la tasa de interés es una herramienta de política importante para el Banco de la Reserva Federal. Calcule una función de respuesta al impulso para un aumento de un punto porcentual en la tasa de interés ( drs ). Dibuje un gráfico de los cambios esperados en la oferta monetaria registrada ( dlrm2 ) y el PIB real registrado ( dlrgdp ). ( Sugerencia: incluya la opción response = c (“dlrgdp,” “dlrm2”) en la función irf .) Sea claro acerca de si está ortogonalizando los residuos o haciendo una suposición teórica sobre el orden de respuesta.
13. Bono: es posible que haya notado que Peake y Eshbaugh-Soha (2008) los datos sobre la cobertura televisiva mensual del tema energético se utilizaron como ejemplo para la regresión del recuento en el Cap. 7 y como ejemplo de serie temporal en este capítulo. Brandt y Williams ( 2001) desarrollan un modelo autorregresivo de Poisson (PAR) para los datos de recuento de series de tiempo, y Fogarty y Monogan (2014) aplican este modelo a estos datos de política energética. Replica este modelo PAR en estos datos. Para obtener información sobre la replicación, consulte: <http://hdl.handle.net/1902.1/16677> .

Notas al pie 1 . Muchos usan modelos ARIMA para pronosticar valores futuros de una serie. Los modelos ARIMA en sí mismos son ateóricos, pero a menudo pueden ser efectivos para la predicción. Dado que la mayor parte del trabajo de ciencia política implica probar hipótesis motivadas teóricamente, esta sección se centra más en el papel que los modelos ARIMA pueden servir para establecer modelos inferenciales.

2 . Si aún no tiene el archivo de datos PESenergy.csv , puede descargarlo del Dataverse (consulte la página vii) o del contenido del capítulo en línea (consulte la página 155).

3 . Además de examinar la serie original o la función de autocorrelación, una prueba de Dickey-Fuller aumentada también sirve para diagnosticar si una serie de tiempo tiene una raíz unitaria. Cargando el URBANA paquete, el comando adf.test llevará a cabo esta prueba en R .

4 . El principal cambio notable es que la versión predeterminada de acf grafica la correlación de retardo cero, ACF (0), que siempre es 1.0. La versión TSA elimina esto y comienza con la primera autocorrelación de retardo, ACF (1).

5 . La fórmula para estas bandas de error es: 0 ± 1. 96 × se r . El error estándar de un coeficiente de correlación es: smir=1 -r2n - 2—-√ . Entonces, en este caso, establecemos r  = 0 bajo la hipótesis nula, y n es el tamaño de la muestra (o la longitud de la serie).

6 . Técnicamente, el PACF en el tercer rezago es negativo y significativo, pero los patrones comunes de los procesos de error sugieren que es poco probable que esto sea una parte crítica del proceso ARIMA.

7 . Aquí mostramos en el texto principal cómo recopilar un diagnóstico a la vez, pero el lector también puede intentar escribir tsdiag (ar1.mod, 24) para recopilar representaciones gráficas de algunos diagnósticos a la vez.

8 . En este caso, tenemos una entrada de pulso, por lo que podemos decir que en noviembre de 1973, el efecto del discurso fue un aumento esperado de 161 noticias, manteniendo todo lo demás igual. En diciembre de 1973, el efecto de arrastre es que esperamos 98 historias más, manteniendo todo lo demás igual porque 161 × 0. 61 ≈ 98. En enero de 1974, el efecto de la intervención es que esperamos 60 historias más, ceteris paribus porque 161 × 0 61 × 0. 61 ≈ 60. El efecto de la intervención continúa hacia adelante en un patrón similar de decadencia. Por el contrario, si hubiéramos obtenido estos resultados con una intervención escalonada en lugar de un pulsointervención, entonces estos efectos se acumularían en lugar de decaer. Bajo este hipotético, los efectos serían 161 en noviembre de 1973, 259 en diciembre de 1973 (porque 161 + 98 = 259) y 319 en enero de 1974 (porque 161 + 98 + 60 = 319).

9 . En particular, en cada etapa del proceso iterativo, el modelo lineal se estima regresando y∗t=yt- ρyt - 1 en X∗t=Xt- ρXt - 1 (Hamilton 1994, pag. 223). Este procedimiento asume que el proceso de ajuste dinámico es el mismo para el resultado y las variables de entrada, lo cual es poco probable. Por tanto, una especificación dinámica como un modelo de rezago distributivo autorregresivo sería más flexible.

10 . Este ejemplo requiere el archivo levant.dta. Descargue este archivo del Dataverse (consulte la página vii) o del contenido en línea de este capítulo (consulte la página 155).

11 . Se le anima a examinar los modelos que habrían sido elegidos por el criterio de Hannan-Quinn (4 rezagos) o el criterio de Schwarz (1 rezago) por su cuenta. ¿Cómo funcionan estos modelos en términos de diagnóstico? ¿Cómo cambiarían las inferencias?

12 . Tenga en cuenta que, de forma predeterminada, el gráfico que presenta R en realidad incluye la correlación perfecta de retraso cero. Si desea eliminar eso, dada nuestra gran longitud de retraso y el tamaño del panel, simplemente cargue el paquete TSA antes de dibujar el gráfico para cambiar el valor predeterminado.

13 . Tenga en cuenta que los intervalos de confianza basados ​​en bootstrap no siempre brindan las coberturas correctas porque confunden la información sobre qué tan bien se ajusta el modelo con la incertidumbre de los parámetros. Por esta razón, los enfoques bayesianos suelen ser la mejor manera de representar la incertidumbre (Brandt y Freeman 2006; Sims y Zha 1999).

14 . Mi agradecimiento a Dave Armstrong por escribir y sugerir este código alternativo.

Material suplementario 318886\_1\_En\_9\_MOESM1\_ESM.zip (400 kb) Dataverse (2,154 KB) Referencias Cuadro GEP, Tiao GC (1975) Análisis de intervenciones con aplicaciones a problemas económicos y ambientales. Asociación J Am Stat 70: 70–79 CrossRefMathSciNetzbMATHGoogle Académico Box GEP, Jenkins GM, Reinsel GC (2008) Análisis de series de tiempo: previsión y control, 4ª ed. Wiley, Hoboken, Nueva Jersey CrossRefzbMATHGoogle Académico Box-Steffensmeier JM, Freeman JR, Hitt MP, Pevehouse JCW (2014) Análisis de series de tiempo para las ciencias sociales. Cambridge University Press, Nueva York CrossRefGoogle Académico Brandt PT, Freeman JR (2006) Avances en el modelado de series de tiempo bayesiano y el estudio de la política: pruebas teóricas, pronósticos y análisis de políticas. Polit Anal 14 (1): 1–36 CrossRefGoogle Académico Brandt PT, Williams JT (2001) Un modelo lineal autorregresivo de Poisson: el modelo de Poisson AR (p). Polit Anal 9 (2): 164–184 CrossRefGoogle Académico Brandt PT, Williams JT (2007) Múltiples modelos de series temporales. Sage, Thousand Oaks, CA Google Académico Cowpertwait PSP, Metcalfe AV (2009) Serie temporal introductoria con R. Springer, Nueva York zbMATHGoogle Académico Cryer JD, Chan KS (2008) Análisis de series de tiempo con aplicaciones en R, 2ª ed. Springer, Nueva York CrossRefzbMATHGoogle Académico Enders W (2009) Series de tiempo econométricas aplicadas, 3ª ed. Wiley, Nueva York Google Académico Fogarty BJ, Monogan JE III (2014) Modelado de datos de recuento de series de tiempo: los desafíos únicos que enfrentan los estudios de comunicación política. Soc Sci Res 45: 73–88 CrossRefGoogle Académico Granger CWJ (1969) Investigación de relaciones causales mediante modelos econométricos y métodos espectrales cruzados. Econometrica 37: 424–438 CrossRefGoogle Académico Granger CWJ, Newbold P (1974) Regresiones espurias en econometría. J Econ 26: 1045–1066 zbMATHGoogle Académico Hamilton JD (1994) Análisis de series de tiempo. Prensa de la Universidad de Princeton, Princeton, Nueva Jersey zbMATHGoogle Académico Keele L, Kelly NJ (2006) Modelos dinámicos para teorías dinámicas: los entresijos de las variables dependientes rezagadas. Polit Anal 14 (2): 186–205 CrossRefGoogle Académico Koyck LM (1954) Rezagos distribuidos y análisis de inversiones. Holanda Septentrional, Amsterdam Google Académico Lütkepohl H (2005) Nueva introducción al análisis de múltiples series de tiempo. Springer, Nueva York CrossRefzbMATHGoogle Académico Mátyás L, Sevestre P (eds) (2008) La econometría de los datos de panel: fundamentos y desarrollos recientes en la teoría y la práctica, 3ª ed. Springer, Nueva York zbMATHGoogle Académico Peake JS, Eshbaugh-Soha M (2008) El impacto en el establecimiento de la agenda de los principales discursos televisivos presidenciales. Polit Commun 25: 113-137 CrossRefGoogle Académico Petris G, Petrone S, Campagnoli P (2009) Modelos lineales dinámicos con R. Springer, Nueva York CrossRefzbMATHGoogle Académico Pfaff B (2008) Análisis de series temporales integradas y cointegradas con R, 2ª ed. Springer, Nueva York CrossRefzbMATHGoogle Académico Shumway RH, Stoffer DS (2006) Análisis de series de tiempo y sus aplicaciones con ejemplos de R, 2ª ed. Springer, Nueva York zbMATHGoogle Académico Sims CA, Zha T (1999) Bandas de error para respuestas de impulso. Econometrica 67 (5): 1113–1155 CrossRefMathSciNetzbMATHGoogle Académico Wakiyama T, Zusman E, Monogan JE III (2014) ¿Puede sostenerse una transición energética baja en carbono en el Japón posterior a Fukushima? Evaluación de los diferentes impactos de los choques exógenos. Política energética 73: 654–666 CrossRefGoogle Académico Wei WWS (2006) Análisis de series de tiempo: métodos univariados y multivariados, 2ª ed. Pearson, Nueva York zbMATHGoogle Académico

# 10 Álgebra lineal con aplicaciones de programación

Palabras clave: - Mínimos cuadrados ordinarios - Multiplicación de matrices - Álgebra de matrices - Marco de datos - Compartir votos

El lenguaje R tiene varios comandos de álgebra matricial incorporados. Esto resulta útil para los analistas que desean escribir sus propios estimadores o tienen otros problemas en álgebra lineal que desean calcular usando software. En algunos casos, es más fácil aplicar una fórmula para predicciones, errores estándar o alguna otra cantidad directamente en lugar de buscar un programa enlatado para calcular la cantidad, si existe. El álgebra de matrices hace que sea sencillo calcular estas cantidades usted mismo. En este capítulo se presenta la sintaxis y comandos disponibles para la realización de álgebra matricial en R .

El capítulo continúa describiendo primero cómo un usuario puede ingresar datos originales a mano, como un medio para crear vectores, matrices y marcos de datos. Luego presenta varios de los comandos asociados con el álgebra lineal. Finalmente, trabajamos a través de un ejemplo aplicado en el que estimamos un modelo lineal con mínimos cuadrados ordinarios programando nuestro propio estimador.

Como datos de trabajo a lo largo del capítulo, consideramos un ejemplo simple de las elecciones al Congreso de Estados Unidos de 2010. Modelamos la participación del candidato republicano en el voto bipartidista en las elecciones para la Cámara de Representantes ( y ) en 2010. Las variables de entrada son una constante ( x 1 ), la participación de Barack Obama en el voto presidencial bipartidista en 2008 ( x 2 ), y la situación financiera del candidato republicano en relación con el demócrata en cientos de miles de dólares ( x 3 ). Para simplificar, modelamos las nueve carreras de la Cámara en el estado de Tennessee. Los datos se presentan en la Tabla 10.1 . Cuadro 10.1 Datos de las elecciones al Congreso de Tennessee en 2010

Distrito

Republicano ( y )

Constante ( x 1 )

Obama ( x 2 )

Financiamiento ( x 3 )

1

0,808

1

0,290

4.984

2

0,817

1

0.340

5.073

3

0.568

1

0.370

12.620

4

0.571

1

0.340

−6,443

5

0.421

1

0.560

−5,758

6

0,673

1

0.370

15.603

7

0,724

1

0.340

14.148

8

0.590

1

0,430

0.502

9

0,251

1

0,770

−9.048

Nota: Datos de Monogan (2013a)

10.1 Creación de vectores y matrices Como primera tarea, al asignar valores a un vector o matriz, debemos usar el comando de asignación tradicional ( <- ). El comando c c ombines varios elementos componentes en un vector objeto. 1 Entonces, para crear un vector a con los valores específicos de 3, 4 y 5:

a <- c (3,4,5)

Dentro de c , todo lo que tenemos que hacer es separar cada elemento del vector con una coma. Como un ejemplo más interesante, supongamos que quisiéramos ingresar tres de las variables de la tabla 10.1 : la participación republicana en el voto bipartidista en 2010, la participación de Obama en el voto bipartidista en 2008 y la ventaja financiera republicana, todos como vectores. Teclearíamos:

Y <-c (.808, .817, .568, .571, .421, .673, .724, .590, .251)

X2 <-c (.29, .34, .37, .34, .56, .37, .34, .43, .77)

X3 <-c (4.984,5.073,12.620, -6.443, -5.758,15.603,14.148,0.502,

-9.048)

Siempre que ingrese datos en forma vectorial, generalmente debemos asegurarnos de haber ingresado el número correcto de observaciones. El comando de longitud devuelve cuántas observaciones hay en un vector. Para verificar las tres entradas de nuestros vectores, escribimos:

largo); longitud (X2); longitud (X3)

Los tres de nuestros vectores deberían tener una longitud de 9 si se ingresaron correctamente. Observe aquí el uso del punto y coma ( ; ). Si un usuario prefiere colocar varios comandos en una sola línea de texto, el usuario puede separar cada comando con un punto y coma en lugar de poner cada comando en una línea separada. Esto permite que los comandos simples se almacenen de forma más compacta.

Si queremos crear un vector que siga un patrón de repetición , podemos usar el comando rep . Por ejemplo, en nuestro modelo de rendimientos electorales de Tennessee, nuestro término constante es simplemente un vector de 1 de 9 × 1:

X1 <- rep (1, 9)

El primer término dentro de rep es el término que se debe repetir y el segundo término es el número de veces que debe repetirse.

Los vectores secuenciales también son fáciles de crear. Dos puntos ( : ) indicaciones R a enteros lista secuencial desde el inicio hasta el punto de parada. Por ejemplo, para crear un índice para el distrito del Congreso de cada observación, necesitaríamos un vector que contenga valores de 1 a 9:

índice <- c (1: 9)

Como acotación al margen, una orden más general es la siguientes comandos, lo que nos permite definir los intervalos de un ss influencia, así como los valores inicial y final. Por ejemplo, si por alguna razón quisiéramos crear una secuencia de - 2 a 1 en incrementos de 0.25, escribiríamos:

e <- seq (-2, 1, por = 0,25)

Cualquier vector que creamos se puede imprimir en la pantalla simplemente escribiendo el nombre del vector. Por ejemplo, si simplemente escribimos Y en el símbolo del sistema, nuestro vector de participación de votos republicanos se imprime en la salida:

[1] 0,808 0,817 0,568 0,571 0,421 0,673 0,724

[8] 0,590 0,251

En este caso, se imprimen los nueve valores del vector Y. El número impreso entre llaves al comienzo de cada fila ofrece el índice del primer elemento de esa fila. Por ejemplo, la octava observación de Y es el valor 0.590. Para vectores más largos, esto ayuda al usuario a realizar un seguimiento de los índices de los elementos dentro del vector.

10.1.1 Crear matrices Pasando a las matrices, podemos crear un objeto de clase matrix de varias formas posibles. Primero, podríamos usar el comando matrix : en el caso más simple posible, supongamos que quisiéramos crear una matriz con todos los valores iguales. Para crear una matriz b de 4 × 4 con cada valor igual a 3:

b <- matriz (3, ncol = 4, nrow = 4, byrow = FALSE)

La sintaxis del comando de matriz primero llama a los elementos que definirán la matriz: en este caso, enumeramos un solo escalar, por lo que este número se repitió para todas las celdas de la matriz. Alternativamente, podríamos listar un vector en su lugar que incluya suficientes entradas para llenar toda la matriz. Luego, necesitamos especificar el número de columnas ( ncol ) y filas ( nrow ). Por último, el argumento byrow se establece en FALSE de forma predeterminada. Con la configuración FALSE , R llena la matriz columna por columna. En su lugar, un ajuste VERDADERO llena la matriz fila por fila. Como regla, siempre que cree una matriz, escriba el nombre de la matriz ( ben este caso) en la consola de comandos para ver si la forma en que R ingresa los datos coincide con lo que pretendía.

Una segunda opción simple es crear una matriz a partir de vectores que ya se han ingresado. Podemos unir los vectores juntos como c OLUMNA vectores utilizando el cbind mando y como r ow vectores utilizando el rbind comando. Por ejemplo, en nuestro modelo de declaraciones de elecciones de Tennessee, necesitaremos crear una matriz de todas las variables de entrada en la que las variables definen las columnas y las observaciones definen las filas. Como hemos definido nuestros tres vectores variables (y cada vector está ordenado por observación), podemos simplemente crear dicha matriz usando cbind :

X <-cbind (1, X2, X3)

X

El comando cbind trata nuestros vectores como columnas en una matriz. Esto es lo que queremos, ya que una matriz de predictores define filas con observaciones y columnas con variables. El 1 en el comando cbind asegura que todos los elementos de la primera columna sean iguales a la constante 1. (Por supuesto, de la forma en que diseñamos X1 , también podríamos haber incluido ese vector). Al escribir X en la consola, obtenemos el imprimir:

X2 X3

[1,] 1 0,29 4,984

[2,] 1 0,34 5,073

[3,] 1 0,37 12,620

[4,] 1 0,34 -6,443

[5,] 1 0,56 -5,758

[6,] 1 0.37 15.603

[7,] 1 0,34 14,148

[8,] 1 0,43 0,502

[9,] 1 0,77 -9,048

Estos resultados coinciden con los datos que tenemos en la tabla 10.1 , por lo que nuestra matriz de covariables debería estar lista cuando estemos listos para estimar nuestro modelo.

Solo para ilustrar el comando rbind , R fácilmente combinaría los vectores como filas de la siguiente manera:

T <-rbind (1, X2, X3)

T

No formateamos datos como este, pero para ver cómo se ven los resultados, al escribir T en la consola se obtiene la impresión:

[, 1] [, 2] [, 3] [, 4] [, 5] [, 6] [, 7]

1.000 1.000 1.00 1.000 1.000 1.000 1.000

X2 0,290 0,340 0,37 0,340 0,560 0,370 0,340

X3 4.984 5.073 12.62 -6.443 -5.758 15.603 14.148

[, 8] [, 9]

1.000 1.000

X2 0,430 0,770

X3 0,502 -9,048

Por tanto, vemos que cada vector variable forma una fila y cada observación forma una columna. En la impresión, cuando R carece de espacio para imprimir una fila completa en una sola línea, envuelve todas las filas a la vez, presentando las columnas 8 y 9 más adelante.

En tercer lugar, podríamos crear una matriz utilizando subíndices. (Los detalles adicionales sobre los subíndices de vectores y matrices se presentan en la Sección  10.1.3 .) A veces, al crear una matriz, el usuario conocerá todos los valores por adelantado, pero en otras ocasiones el usuario debe crear una matriz y luego completar la valores más tarde. En el último caso, es una buena idea crear “espacios en blanco” para completar designando cada celda como faltante (o NA ). La buena característica de esto es que un usuario puede identificar fácilmente una celda que nunca se completó. Por el contrario, si se crea una matriz con algún valor numérico predeterminado, digamos 0, más adelante es imposible distinguir una celda que tiene un 0 predeterminado de una con un valor real de 0. Entonces, si quisiéramos crear un 3 × 3 matriz denominada en blanco para rellenar más tarde, escribiríamos:

blanco <- matriz (NA, ncol = 3, nrow = 3)

Si luego quisiéramos asignar el valor de 8 a la primera fila, tercer elemento de columna, escribiríamos:

en blanco [1,3] <- 8

Si luego quisiéramos insertar el valor π (= 3. 141592 … ) en la segunda fila, la entrada de la primera columna, escribiríamos:

en blanco [2,1] <- pi

Si quisiéramos usar nuestro vector previamente definido a  = (3, 4, 5) ′ para definir la segunda columna, escribiríamos:

en blanco [, 2] <- a

Luego podríamos verificar nuestro progreso simplemente escribiendo en blanco en el símbolo del sistema, que imprimiría:

[, 1] [, 2] [, 3]

[1,] NA 3 8

[2,] 3,141593 4 NA

[3,] NA 5 NA

A la izquierda de la matriz, se definen los términos de las filas. En la parte superior de la matriz, se definen los términos de la columna. Observe que cuatro elementos todavía se codifican NA porque nunca se ofreció un valor de reemplazo.

En cuarto lugar, a diferencia de rellenar matrices después de la creación, también podemos conocer los valores que queremos en el momento de crear la matriz. Como ejemplo simple, para crear una matriz W de 2 × 2 en la que enumeramos el valor de cada celda columna por columna:

W <- matriz (c (1,2,3,4), ncol = 2, nrow = 2)

Observe que nuestro primer argumento ahora es un vector porque queremos proporcionar elementos únicos para cada una de las celdas de la matriz. Con cuatro elementos en el vector, tenemos el número correcto de entradas para una matriz de 2 × 2. Además, en este caso, ignoramos el argumento byrow porque el valor predeterminado es completar por columnas. Por el contrario, si quisiéramos enumerar los elementos de celda fila por fila en la matriz Z , simplemente estableceríamos byrow en VERDADERO :

Z <- matriz (c (1,2,3,4), ncol = 2, nrow = 2, byrow = TRUE)

Escriba W y Z en la consola y observe cómo se ha reorganizado el mismo vector de entradas de celda pasando de una celda a otra.

Alternativamente, supongamos que queríamos crear un 10 × 10 matriz N dónde estaba cada entrada en la célula un r Andom extraer de una norma de distribución al con media 10 y desviación estándar 2, o norte( 10 , 4 ) :

N <- matriz (rnorm (100, media = 10, sd = 2), nrow = 10, ncol = 10)

Debido a que rnorm devuelve un objeto de la clase de vector, nuevamente estamos listando un vector para crear nuestras entradas de celda. El comando rnorm se extrae de la distribución normal con nuestras especificaciones 100 veces, lo que proporciona el número de entradas de celda que necesitamos para una matriz de 10 × 10.

En quinto lugar, en muchas ocasiones, que se desee crear una matriz diagonal que contiene sólo elementos en su principal diag onal (desde la parte superior izquierda a la parte inferior derecha), con ceros en todas las demás células. El comando diag facilita la creación de dicha matriz:

D <- diag (c (1: 4))

Al escribir D en la consola, ahora vemos cómo aparece esta matriz diagonal:

[, 1] [, 2] [, 3] [, 4]

[1,] 1 0 0 0

[2,] 0 2 0 0

[3,] 0 0 3 0

[4,] 0 0 0 4

Además, si uno inserta una matriz cuadrada en el comando diag , devolverá un vector de los elementos diagonales de la matriz. Por ejemplo, en una matriz de varianza-covarianza, los elementos diagonales son varianzas y se pueden extraer rápidamente de esta manera.

10.1.2 Conversión de matrices y tramas de datos Un medio final de crear un objeto de matriz es con el comando as.matrix . Este comando puede tomar un objeto de la clase de marco de datos y convertirlo en un objeto de la clase de matriz . Para un marco de datos llamado mydata , por ejemplo, el comando mydata.2 <- as.matrix (mydata) coaccionaría los datos en forma de matriz, haciendo que todas las operaciones de matriz subsiguientes fueran aplicables a los datos. Además, escribir mydata.3 <- as.matrix (mydata [, 4: 8]) solo tomaría las columnas de la cuatro a la ocho del marco de datos y las convertiría en una matriz, lo que permitiría al usuario dividir los datos en subconjuntos mientras crea un objeto de matriz .

De manera similar, suponga que queremos tomar un objeto de la clase de matriz y crear un objeto de la clase de marco de datos , tal vez nuestros datos electorales de Tennessee de la tabla 10.1 . En este caso, el comando as.data.frame funcionará. Si simplemente quisiéramos crear un marco de datos a partir de nuestra matriz de covariables X , podríamos escribir:

X.df <- como.data.frame (X)

Si quisiéramos crear un marco de datos utilizando todas las variables de la Tabla 10.1 , incluida la variable dependiente y el índice, podríamos escribir:

tennessee <- como.data.frame (cbind (index, Y, X1, X2, X3))

En este caso, el comando cbind incrustado en as.data.frame define una matriz, que se convierte inmediatamente en un marco de datos. En general, entonces, si un usuario desea ingresar sus propios datos en R , la forma más sencilla será definir vectores variables, unirlos como columnas y convertir la matriz en un marco de datos.

10.1.3 Subíndice Como se mencionó en la sección de creación de matrices, llamar a elementos de vectores y matrices por sus subíndices puede ser útil para extraer la información necesaria o para realizar asignaciones. Para llamar a un valor específico, podemos indexar un vector y por el n- ésimo elemento, usando Y [n] . Entonces, el tercer elemento del vector y , o y 3 , es llamado por:

Y [3]

Para indexar una matriz X n × k para el valor X ij , donde i representa la fila y j representa la columna, use la sintaxis X [i, j] . Si queremos seleccionar todos los valores de la j- ésima columna de X , podemos usar X [, j] . Por ejemplo, para devolver la segunda columna de la matriz X , escriba:

X [, 2]

Alternativamente, si una columna tiene un nombre, entonces el nombre (entre comillas) también se puede usar para llamar a la columna. Por ejemplo:

X [, “X2”]

De manera similar, si deseamos seleccionar todos los elementos de la i- ésima fila de X , podemos usar X [i,] . Para la primera fila de la matriz X , escriba:

X [1,]

Alternativamente, también podríamos usar un nombre de fila, aunque la matriz X no tiene nombres asignados a las filas.

Sin embargo, si lo deseamos, podemos crear nombres de filas para matrices. Por ejemplo:

nombres de fila (X) <- c (“Dist. 1,” “Dist. 2,” “Dist. 3,” “Dist. 4,”

"Dist. 5", "Dist. 6", "Dist. 7", "Dist. 8", "Dist. 9")

Del mismo modo, el comando COLNAMES permite al usuario definir col UMN nombres . Para simplemente escribir nombres de fila (X) o colnames (X) sin realizar una asignación, R imprimirá los nombres de fila o columna guardados para una matriz.

10.2 Comandos vectoriales y matriciales Ahora que tenemos la sensación de crear vectores y matrices, recurrimos a comandos que extraen información de estos objetos o nos permiten realizar álgebra lineal con ellos. Como se mencionó anteriormente, después de ingresar un vector o matriz en R , además de imprimir el objeto en la pantalla para una inspección visual, también es una buena idea verificar y asegurarse de que las dimensiones del objeto sean correctas. Por ejemplo, para obtener la longitud de un vector a :

longitud (a)

Del mismo modo, para obtener las dim ensiones de una matriz, tipo:

tenue (X)

El comando dim primero imprime el número de filas, luego el número de columnas. La verificación de estas dimensiones ofrece una garantía adicional de que los datos de una matriz se han introducido correctamente.

En el caso de los vectores, los elementos se pueden tratar como datos y se pueden extraer cantidades resumidas. Por ejemplo, para sumar la suma de los elementos de un vector:

suma (X2)

De manera similar, para tomar la media de los elementos de un vector (en este ejemplo, la media del voto por distrito de Obama en 2008):

media (X2)

Y tomar la var iance de un vector:

var (X2)

Otra opción que tenemos para matrices y vectores es que podemos tomar muestras de un objeto dado con el comando sample . Supongamos que queremos una muestra de diez números de N , nuestra matriz de 10 × 10 de extracciones aleatorias de una distribución normal:

set.seed (271828183)

N <- matriz (rnorm (100, media = 10, sd = 2), nrow = 10, ncol = 10)

s <- muestra (N, 10)

El primer comando, set.seed , hace que esta simulación sea más replicable. (Ver. Cap  11 para obtener más detalles acerca de este comando.) Esto nos da un vector llamado s de elementos aleatorios de diez N . También tenemos la opción de aplicar el comando de ejemplo a los vectores. 2

El comando de aplicación es a menudo la forma más eficiente de hacer cálculos vectorizados. Por ejemplo, para calcular las medias de todas las columnas en nuestra matriz de datos de Tennessee X :

aplicar (X, 2, media)

En este caso, el primer argumento enumera la matriz a analizar. El segundo argumento, 2 , le dice a R que aplique una función matemática a lo largo de las columnas de la matriz. El tercer argumento, mean , es la función que queremos aplicar a cada columna. Si quisiéramos la media de las filas en lugar de las columnas, podríamos usar un 1 como segundo argumento en lugar de un 2. Cualquier función definida en R se puede usar con el comando apply .

10.2.1 Álgebra de matrices También están disponibles los comandos que son más específicos del álgebra matricial. Siempre que el usuario desee guardar la salida de una operación vectorial o matricial, se debe utilizar el comando de asignación. Por ejemplo, si por alguna razón necesitáramos la diferencia entre cada participación republicana del voto bipartidista y la participación de Obama en el voto bipartidista, podríamos asignar el vector m a la diferencia de los dos vectores escribiendo:

m <- Y - X2

Con las operaciones aritméticas, R es bastante flexible, pero una palabra sobre cómo funcionan los comandos puede evitar confusiones futuras. Para suma ( + ) y resta ( - ): si los argumentos son dos vectores de la misma longitud (como es el caso en el cálculo del vector m ), entonces R calcula la suma o resta de vectores regulares donde cada elemento se suma o se resta de su elemento correspondiente en el otro vector. Si los argumentos son dos matrices del mismo tamaño, se aplica la suma o resta de matrices cuando cada entrada se combina con su entrada correspondiente en la otra matriz. Si un argumento es un escalar, entonces el escalar se agregará a cada elemento del vector o matriz.

Nota : Si se agregan dos vectores de longitud desigual o dos matrices de diferente tamaño, aparece un mensaje de error, ya que estos argumentos no son conformes. Para ilustrar esto, si intentáramos sumar nuestra muestra de diez números de antes, s , a nuestro vector de la participación de Obama en 2008 en los votos, x 2 , de la siguiente manera:

s + X2

En este caso recibiríamos una salida que deberíamos ignorar, seguida de un mensaje de error en rojo:

[1] 11.743450 8.307068 11.438161 14.251645 10.828459

[6] 10.336895 9.900118 10.092051 12.556688 9.775185

Mensaje de advertencia:

En s + X2: la longitud del objeto más larga no es múltiplo de más corta

longitud del objeto

Este mensaje de advertencia nos dice que la salida es una tontería. Dado que el vector s tiene una longitud de 10, mientras que el vector X2 tiene una longitud de 9, lo que R ha hecho aquí es agregar el décimo elemento de sa el primer elemento de X2 y usarlo como el décimo elemento de la salida. El mensaje de error que sigue sirve como recordatorio de que una entrada basura que rompe las reglas del álgebra lineal produce una salida basura que nadie debería usar.

Por el contrario, \* , / , ^ (exponentes), exp (función exponencial) y log aplican la operación relevante a cada entrada escalar en el vector o matriz. Para nuestra matriz de predictores de Tennessee, por ejemplo, pruebe el comando:

X.sq <- X ^ 2

Esto volvería una matriz que los cuadrados de cada celda por separado en la matriz X . De manera similar, la operación de multiplicación simple ( \* ) realiza la multiplicación elemento por elemento. Por lo tanto, si dos vectores son de la misma longitud o dos matrices del mismo tamaño, la salida será un vector o matriz del mismo tamaño donde cada elemento es el producto de los elementos correspondientes. A modo de ilustración, multipliquemos la participación de Obama en el voto bipartidista por la ventaja financiera republicana de varias formas. (Las cantidades pueden ser tontas, pero esto ofrece un ejemplo de cómo funciona el código). Intente, por ejemplo:

x2.x3 <- X2 \* X3

El vector de salida es:

[1] 1.44536 1.72482 4.66940 -2.19062 -3.22448

[6] 5.77311 4.81032 0.21586 -6.96696

Esto corresponde a la multiplicación escalar elemento por elemento.

Más a menudo, el usuario realmente querrá realizar una multiplicación de matrices adecuada. En R , la multiplicación de matrices se realiza mediante la operación % \*% . Entonces, si hubiéramos multiplicado nuestros dos vectores de participación de votos de Obama y ventaja financiera republicana de esta manera:

x2.x3.inner <- X2% \*% X3

R ahora devolvería el producto interno o el producto escalar ( X2⋅X3 ). Esto equivale a 6.25681 en nuestro ejemplo. Otra cantidad útil para la multiplicación de vectores es el producto externo o producto tensorial ( X2⊗X3 ). Podemos obtener esto transponiendo el segundo vector en nuestro código:

x2.x3.outer <- X2% \*% t (X3)

La salida de este comando es una matriz de 9 × 9. Para obtener esta cantidad se utilizó el comando t ranspose ( t ). 3 En el álgebra de matrices, es posible que a su vez vectores fila a vectores de columna o viceversa, y el t operación ranspose logra esto. De manera similar, cuando se aplica a una matriz, cada fila se convierte en una columna y cada columna se convierte en una fila.

Como un ejemplo más de multiplicación de matrices, la variable de entrada matriz X tiene un tamaño de 9 × 3, y la matriz T que permite que las variables definan filas es una matriz de 3 × 9. En este caso, se podría crear la matriz P  =  TX porque el número de columnas de T es el mismo que el número de filas de X . Por lo tanto, podemos decir que las matrices son compatibles para la multiplicación y darán como resultado una matriz de tamaño 3 × 3. Podemos calcular esto como:

P <- T% \*% X

Tenga en cuenta que si nuestras matrices no son aptas para la multiplicación, R devolverá un mensaje de error. 4

Más allá de la multiplicación de matrices y la operación de transposición, otra cantidad importante que es exclusiva del álgebra de matrices es el det erminante de una matriz cuadrada (o una matriz que tiene el mismo número de filas que de columnas). Nuestra matriz P es cuadrada porque tiene tres filas y tres columnas. Una razón para calcular el determinante es que una matriz cuadrada tiene una inversa solo si el determinante es distinto de cero. 5 Para calcular el determinante de P , escribimos:

det (P)

Esto arroja un valor de 691,3339. Por tanto, sabemos que P tiene una inversa.

Para una matriz cuadrada que se puede invertir, la inversa es una matriz que se puede multiplicar por la original para producir la matriz identidad. Con la solución de comandos, R o bien resolver por la inversa de la matriz o transmitir que la matriz es invertible. Para probar este concepto, escriba:

invP <- resolver (P)

invP% \*% P

En la primera línea, creamos P -1 , que es la inversa de P . En la segunda línea, multiplicamos P −1 P y la impresión es:

X2 X3

1.000000e + 00 -6.106227e-16 -6.394885e-14

X2 1.421085e-14 1.000000e + 00 1.278977e-13

X3 2.775558e-17 -2.255141e-17 1.000000e + 00

Esta es la forma básica de la matriz de identidad, con valores de 1 a lo largo de la diagonal principal (desde la parte superior izquierda a la inferior derecha) y valores de 0 fuera de la diagonal. Mientras que los elementos fuera de la diagonal no se enumeran exactamente como cero, esto puede atribuirse a error de redondeo en R parte ‘s. La notación científica para la segunda fila, el primer elemento de la columna, por ejemplo, significa que los primeros 13 dígitos después del lugar decimal son cero, seguidos de un 1 en el decimocuarto dígito.

10.3 Ejemplo aplicado: Programación de regresión OLS Para ilustrar las diversas operaciones de álgebra matricial que R tiene disponibles, en esta sección trabajaremos un ejemplo aplicado calculando el estimador de mínimos cuadrados ordinarios (MCO) con nuestro propio programa usando datos reales.

10.3.1 Cálculo manual de OLS Primero, para motivar el trasfondo del problema, considere la formulación del modelo y cómo lo estimaríamos a mano. Nuestro modelo de regresión lineal poblacional para la participación de votos en Tennessee es: y = X β+ u . En este modelo, y es un vector de la proporción de votos republicanos en cada distrito, X es una matriz de predictores (incluida una constante, la proporción de votos de Obama en 2008 y la ventaja financiera del republicano en relación con la demócrata), β consta del coeficiente parcial para cada predictor, y u es un vector de perturbaciones. Nosotros estimamos β^= (X′X)- 1X′y , produciendo la función de regresión muestral y^= Xβ^ .

Para iniciar el cálculo de la mano, tenemos que definir X . Tenga en cuenta que debemos incluir un vector de 1 para estimar un término de intersección. En forma escalar, nuestro modelo de población es: yI=β1X1 yo+β2X2 yo+β3X3 yo+tuI , donde x 1 i  = 1 para todo i . Esto nos da la matriz de predictores: X =⎡⎣⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢1111111110,290,340,370,340,560,370,340,430,774.9845.07312.620- 6.443- 5.75815.60314.1480.502- 9.048⎤⎦⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥ A continuación, premultiplica X por su transposición: X′X =⎡⎣⎢1.0000,2904.9841.0000.3405.0731.0000.37012.6201.0000.340- 6.4431.0000.560- 5.7581.0000.37015.6031.0000.34014.1481.0000,4300.5021.0000,770- 9.048⎤⎦⎥⎡⎣⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢1111111110,290,340,370,340,560,370,340,430,774.9845.07312.620- 6.443- 5.75815.60314.1480.502- 9.048⎤⎦⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥ que funciona para: X′X =⎡⎣⎢9.000003.8100031.681003.810001.796106.2568131.681006.25681810.24462⎤⎦⎥ También necesitamos X′y : X′y =⎡⎣⎢1.0000,2904.9841.0000.3405.0731.0000.37012.6201.0000.340- 6.4431.0000.560- 5.7581.0000.37015.6031.0000.34014.1481.0000,4300.5021.0000,770- 9.048⎤⎦⎥⎡⎣⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢⎢0,8080,8170.5680.5710.4210,6730,7240.5900,251⎤⎦⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥⎥ que funciona para: X′y =⎡⎣⎢5.42302.094328.0059⎤⎦⎥ Invertir a mano La última cantidad que necesitamos es la inversa de X′X . Haciendo esto a mano, podemos resolver por eliminación de Gauss-Jordan: [X′X | Yo ]=⎡⎣⎢⎢9.000003.8100031.681003.810001.796106.2568131.681006.25681810.244621,000000,000000,000000,000001,000000,000000,000000,000001,00000⎤⎦⎥⎥ Dividir la fila 1 entre 9: ⎡⎣⎢⎢1,000003.8100031.681000.423331.796106.256813.520116.25681810.244620.111110,000000,000000,000001,000000,000000,000000,000001,00000⎤⎦⎥⎥ Reste 3,81 multiplicado por la fila 1 de la fila 2: ⎡⎣⎢⎢1,000000,0000031.681000.423330.183206.256813.52011- 7.15481810.244620.11111- 0,423330,000000,000001,000000,000000,000000,000001,00000⎤⎦⎥⎥ Reste 31.681 veces la fila 1 de la fila 3: ⎡⎣⎢⎢1,000000,000000,000000.423330.18320- 7.154813.52011- 7.15481698.724020.11111- 0,42333- 3.520110,000001,000000,000000,000000,000001,00000⎤⎦⎥⎥ Divida la fila 2 entre 0,1832: ⎡⎣⎢⎢1,000000,000000,000000.423331,00000- 7.154813.52011- 39.05464698.724020.11111- 2.31077- 3.520110,000005.458520,000000,000000,000001,00000⎤⎦⎥⎥ Suma 7,15481 veces la fila 2 a la fila 3: ⎡⎣⎢⎢1,000000,000000,000000.423331,000000,000003.52011- 39.05464419.295490.11111- 2.31077- 20.053230,000005.4585239.054670,000000,000001,00000⎤⎦⎥⎥ Dividir la fila 3 entre 419.29549: ⎡⎣⎢⎢1,000000,000000,000000.423331,000000,000003.52011- 39.054641,000000.11111- 2.31077- 0.047830,000005.458520.093140,000000,000000,00239⎤⎦⎥⎥ Sumar 39.05464 veces la fila 3 a la fila 2: ⎡⎣⎢⎢1,000000,000000,000000.423331,000000,000003.520110,000001,000000.11111- 4.17859- 0.047830,000009.096070.093140,000000.093340,00239⎤⎦⎥⎥ Reste 3,52011 multiplicado por la fila 3 de la fila 1: ⎡⎣⎢⎢1,000000,000000,000000.423331,000000,000000,000000,000001,000000.27946- 4.17859- 0.04783- 0.327869.096070.09314- 0,008410.093340,00239⎤⎦⎥⎥ Reste 42333 multiplicado por la fila 2 de la fila 1: ⎡⎣⎢⎢1,000000,000000,000000,000001,000000,000000,000000,000001,000002.04838- 4.17859- 0.04783- 4.178499.096070.09314- 0.047920.093340,00239⎤⎦⎥⎥= [ I | (X′X)- 1] Como una pequeña arruga en estos cálculos manuales, podemos ver que ( X′X ) −1 está un poco fuera de lugar debido a un error de redondeo. En realidad, debería ser una matriz simétrica. Respuesta final a mano Si postmultiplicamos ( X′X ) −1 por X′y obtenemos: (X′X)- 1X′y =⎡⎣⎢2.04838- 4.17859- 0.04783- 4.178499.096070.09314- 0.047920.093340,00239⎤⎦⎥⎡⎣⎢5.42302.094328.0059⎤⎦⎥=⎡⎣⎢1.0178- 1,00180,0025⎤⎦⎥=β^ O en forma escalar: y^I= 1.0178 - 1.0018X2 yo+ 0,0025X3 yo . 10.3.2 Escribir un estimador de MCO en R Dado que invertir la matriz requirió tantos pasos e incluso se produjo algún error de redondeo, será más fácil que R haga algo del trabajo pesado por nosotros. (A medida que aumente el número de observaciones o variables, encontraremos la ayuda computacional aún más valiosa). Para programar nuestro propio estimador de MCO en R , los comandos clave que necesitamos son: Multiplicación de matrices: % \*%

Transponer: t

Matriz inversa: resolver

Sabiendo esto, es fácil programar un estimador para β^= (X′X)- 1X′y .

Primero debemos ingresar nuestros vectores variables usando los datos de la tabla 10.1 y combinar las variables de entrada en una matriz. Si aún no los ha introducido, escriba:

Y <-c (.808, .817, .568, .571, .421, .673, .724, .590, .251)

X1 <- rep (1, 9)

X2 <-c (.29, .34, .37, .34, .56, .37, .34, .43, .77)

X3 <-c (4.984,5.073,12.620, -6.443, -5.758,15.603,14.148,0.502,

-9.048)

X <-cbind (X1, X2, X3)

Para estimar MCO con nuestro propio programa, simplemente necesitamos traducir el estimador ( X′X ) −1 X′y a la sintaxis R :

beta.hat <-solve (t (X)% *% X)%* % t (X)% \*% Y

beta.hat

Desglosando esto un poco: el comando solve comienza porque la primera cantidad es inversa, ( X′X ) −1 . Dentro de la llamada a resolver , el primer argumento debe transponerse (de ahí el uso de t ) y luego se posmultiplica por la matriz de covariables no transpuesta (de ahí el uso de % \*% ). Seguimos postmultiplicando la transpuesta de X , luego postmultiplicando el vector de resultados ( y ). Cuando imprimimos nuestros resultados, obtenemos:

[, 1]  
  
1.017845630

X2 -1.001809341

X3 0,002502538

Estos son los mismos resultados que obtuvimos a mano, a pesar de las discrepancias de redondeo que encontramos al invertir por nuestra cuenta. Sin embargo, escribir el programa en R nos dio al mismo tiempo un control total sobre el estimador, siendo mucho más rápido que las operaciones manuales.

Por supuesto, una opción aún más rápida sería usar el comando enlatado lm que usamos en el Cap. 6 :

tennessee <- como.data.frame (cbind (Y, X2, X3))

lm (Y ~ X2 + X3, datos = Tennessee)

Esto también produce exactamente los mismos resultados. En la práctica, siempre que se calculan las estimaciones de MCO, este es casi siempre el enfoque que queremos adoptar. Sin embargo, este procedimiento nos ha permitido verificar la utilidad de los comandos de álgebra matricial de R. Si el usuario encuentra la necesidad de programar un estimador más complejo para el cual no existe un comando predefinido, éste debería ofrecer herramientas relevantes para lograrlo.

10.3.3 Otras aplicaciones Con estas herramientas básicas en la mano, los usuarios ahora deberían poder comenzar a programar con matrices y vectores. Algunas aplicaciones intermedias de esto incluyen: calcular errores estándar a partir de modelos estimados con optim (ver capítulo  11 ) y valores predichos para formas funcionales complicadas (ver el código que produjo la figura  9.3 ). Algunas de las aplicaciones más avanzadas que pueden beneficiarse del uso de estas herramientas incluyen: mínimos cuadrados generalizados factibles (incluidos los mínimos cuadrados ponderados), optimización de funciones de verosimilitud para distribuciones normales multivariadas y generación de datos correlacionados mediante la descomposición de Chol esky (consulte el comando chol ). Muy pocos programas ofrecen la flexibilidad de estimación y programación que R lo hace siempre que el álgebra matricial es esencial para el proceso.

10.4 Problemas de práctica Para estos problemas de práctica, considere los datos sobre las elecciones al Congreso en Arizona en 2010, que se presentan a continuación en la Tabla 10.2 : Cuadro 10.2 Datos de las elecciones al Congreso de Arizona en 2010

Distrito

Republicano ( y )

Constante ( x 1 )

Obama ( x 2 )

Financiamiento ( x 3 )

1

0,50

1

0,44

−9,11

2

0,65

1

0,38

8.31

3

0,52

1

0,42

8.00

4

0,28

1

0,66

−8,50

5

0,52

1

0,47

−7,17

6

0,67

1

0,38

5.10

7

0,45

1

0,57

−5,32

8

0,47

1

0,46

−18,64

Nota: Datos de Monogan (2013a)

1. Cree un vector que consista en la proporción republicana del voto bipartidista ( y ) en los ocho distritos electorales de Arizona. Usando cualquier medio que prefiera, cree una matriz, X , en la que las ocho filas representan los ocho distritos de Arizona, y las tres columnas representan un vector constante de unos ( x 1 ), la participación de Obama en 2008 en los votos ( x 2 ) y el Saldo financiero republicano ( x 3 ). Imprima su vector y matriz, y pídale a R que calcule la longitud del vector y las dimensiones de la matriz. ¿Todos estos resultados son consistentes con los datos que ve en la Tabla 10.2 ?
2. Usando su matriz, X , calcule X’X usando los comandos de R. Cual es tu resultado?
3. Usando los comandos de R , encuentre la inversa de su respuesta anterior. Es decir, calcule ( X′X ) −1 . Cual es tu resultado?
4. Con los datos de la tabla 10.2 , considere el modelo de regresión yI=β1X1 yo+β2X2 yo+β3X3 yo+tuI . Usando los comandos de álgebra matricial de R , calcule β^= (X′X)- 1X′y . ¿Cuáles son sus estimaciones de los coeficientes de regresión? β^1 , β^2 , y β^3 ? ¿Cómo se comparan sus resultados si los calcula usando el comando lm del Cap. 6 ?

Notas al pie 1 . Alternativamente, cuando los elementos combinados son objetos complejos, c crea en su lugar un objeto de lista .

2 . Para los lectores interesados ​​en bootstrapping, que es una de las aplicaciones más comunes de muestreo a partir de datos, el enfoque más eficiente será instalar el paquete de arranque y probar algunos de los ejemplos que ofrece la biblioteca.

3 . Una sintaxis alternativa habría sido X2% o% X3 .

4 . Una variedad única de multiplicación de matrices se denomina producto de Kronecker ( H ⊗ L ). El producto Kronecker tiene aplicaciones útiles en el análisis de datos de panel. Consulte el comando kronecker en R para obtener más información.

5 . Como otra aplicación en estadística, la función de verosimilitud para una distribución normal multivariante también recurre al determinante de la matriz de covarianza.

Material suplementario 318886\_1\_En\_10\_MOESM1\_ESM.zip (106 kb) Dataverse (2,154 KB) Referencias Monogan JE III (2013a) Un caso para registrar estudios de resultados políticos: una aplicación en las elecciones a la Cámara de 2010. Polit Anal 21 (1): 21–37 CrossRefGoogle Académico

# 11 Herramientas de programación adicionales

Palabras clave: - Estimación de máxima verosimilitud - Votante promedio - Análisis de Monte Carlo - Experimento de Montecarlo - Posición de emisión

Como se ha mostrado en los últimos capítulos, R ofrece a los usuarios flexibilidad y oportunidades para análisis de datos avanzados que no se encuentran en muchos programas. En este capítulo final, exploraremos las herramientas de programación de R , que permiten al usuario crear código que aborde cualquier problema único que enfrente.

Tenga en cuenta que muchas de las herramientas relevantes para la programación ya se han presentado anteriormente en este libro. En el Cap. 10 , se introdujeron las herramientas para el álgebra matricial en R y muchos programas requieren procesamiento de matrices. Además, las declaraciones lógicas (o booleanas) son esenciales para la programación. Los operadores lógicos de R se introdujeron en el Cap. 2 , por lo tanto, consulte la Tabla 2.1 para recordar cuál es la función de cada operador.

Las secciones siguientes presentarán varias otras herramientas que son importantes para la programación: distribuciones de probabilidad, definición de nuevas funciones, bucles, ramificaciones y optimización (que es particularmente útil para la estimación de máxima verosimilitud). El capítulo terminará con dos grandes ejemplos aplicados. El primero, basado en Monogan (2013b), introduce la programación orientada a objetos en R y aplica varias herramientas de programación de este capítulo para encontrar soluciones a un problema de teoría de juegos insoluble. El segundo, dibujo de Signorino (1999), Es un ejemplo de análisis de Monte Carlo y la estimación de máxima verosimilitud más avanzada en R . 1 Juntas, las dos aplicaciones deben mostrar cómo todas las herramientas de programación pueden unirse para resolver un problema complejo.

11.1 Distribuciones de probabilidad R le permite utilizar una amplia variedad de distribuciones para cuatro propósitos. Para cada distribución, R le permite llamar a la función de distribución acumulativa (CDF), la función de densidad de probabilidad (PDF), la función de cuantiles y las extracciones aleatorias de la distribución. Todos los comandos de distribución de probabilidad constan de un prefijo y un sufijo. La tabla 11.1 presenta los cuatro prefijos y su uso, así como los sufijos para algunas distribuciones de probabilidad de uso común. Las funciones de cada distribución toman argumentos únicos para los parámetros de esa distribución de probabilidad. Para ver cómo éstos se especifican, los archivos de uso de ayuda (por ejemplo, ? Punif , ? PEXP , o ? Pnorm ). 2 Cuadro 11.1 Usando distribuciones de probabilidad en R

Prefijo

Uso

Sufijo

Distribución

pag

Función de distribución acumulativa

norma

Normal

D

Función de densidad de probabilidad

logis

Logístico

q

Función cuantil

t

t

r

Sorteo aleatorio de la distribución

F

F

unif

Uniforme

pois

Poisson

Exp

Exponencial

chisq

Chi-cuadrado

binom

Binomio

Si desea conocer la probabilidad de que una observación normal estándar sea menor que 1.645, use el comando pnorm de la función de distribución acumulativa (CDF) :

normal (1.645)

Suponga que desea dibujar un escalar de la distribución normal estándar: para dibujar a ∼ N( 0 , 1 ) , use el comando de dibujo aleatorio rnorm :

a <- normal (1)

Para dibujar un vector con diez valores de una distribución χ 2 con cuatro grados de libertad, use el comando de dibujo aleatorio:

c <- rchisq (10, gl = 4)

Recuerde del Cap. 10 que el comando de ejemplo también nos permite simular valores, siempre que proporcionemos un vector de valores posibles. Por tanto, R ofrece una amplia gama de comandos de simulación de datos.

Suponga que tenemos una probabilidad dada, 0.9, y queremos saber el valor de una distribución χ 2 con cuatro grados de libertad en la cual la probabilidad de ser menor o igual a ese valor es 0.9. Esto requiere la función de cuantiles :

qchisq (.9, gl = 4)

Podemos calcular la probabilidad de un cierto valor a partir de la función de masa de probabilidad (PMF) para una distribución discreta. De una distribución de Poisson con parámetro de intensidad 9, ¿cuál es la probabilidad de un conteo de 5?

dpois (5, lambda = 9)

Aunque suele ser de menor interés, para una distribución continua, podemos calcular el valor de la función de densidad de probabilidad (PDF) de un valor particular. Esto no tiene un significado inherente, pero ocasionalmente es necesario. Para una distribución normal con media 4 y desviación estándar 2, la densidad en el valor 1 viene dada por:

dnorm (1, media = 4, sd = 2)

11.2 Funciones R le permite crear sus propias funciones con el comando de función . El comando de función utiliza la siguiente sintaxis básica:

function.name <- function (ENTRADAS) {BODY}

Observe que el comando de función primero espera que las entradas se enumeren entre paréntesis, mientras que el cuerpo de la función se enumera entre llaves. Como se puede ver, hay pocas restricciones en lo que el usuario elige poner en la función, por lo que una función se puede diseñar para lograr lo que el usuario desee.

Por ejemplo, supongamos que estamos interesados ​​en la ecuación y= 2 +1X2 . Podríamos definir esta función fácilmente en R . Todo lo que tenemos que hacer es especificar que nuestra variable de entrada es x y nuestro cuerpo es el lado derecho de la ecuación. Esto creará una función que devuelve valores para y . Definimos nuestra función, llamada first.fun , de la siguiente manera:

first.fun <-función (x) {

y <-2 + x ^ {- 2}  
  
 retorno (y)  
  
 }

Aunque está dividido en unas pocas líneas, todo esto es un gran comando, ya que las llaves ( {} ) abarcan varias líneas. Con el comando de función , comenzamos declarando que x es el nombre de nuestra única entrada, basado en el hecho de que nuestra ecuación de interés tiene x como el nombre de la variable de entrada. A continuación, dentro de las llaves, asignamos la salida y como la función exacta de x que nos interesa. Como último paso antes de cerrar nuestras llaves, usamos el comando return , que le dice a la función cuál es el resultado de salida después de que se llame. El regresoEl comando es útil para determinar la salida de una función por un par de razones: Primero, si definimos varios objetos dentro de un comando, esto nos permite especificar lo que debería imprimirse en la salida versus lo que era estrictamente interno a la función. En segundo lugar, return nos permite informar una lista de elementos como salida, si lo deseamos. Alternativamente, podríamos haber sustituido el comando invisible , que funciona de la misma manera que return , excepto que no imprime la salida en la pantalla, sino que simplemente almacena la salida.

Con esta función, si queremos saber qué valor toma y cuando x  = 1, solo necesitamos escribir: first.fun (1) . Como usamos return en lugar de invisible , la impresión simplemente dice:

[1] 3

Por supuesto, el resultado de que y  = 3 aquí se puede verificar fácilmente a mano. De manera similar, sabemos que si x  = 3, entonces y= 219 . Para verificar este hecho, podemos escribir: first.fun (3) . R nos dará la impresión correspondiente:

[1] 2.111111

Como tarea final con esta función, podemos graficar cómo se ve insertando un vector de valores x en ella. Considere el código:

my.x <-seq (-4,4, por = .01)

plot (y = first.fun (my.x), x = my.x, type = “l,”

xlab = "x", ylab = "y", ylim = c (0,10))

Esto producirá el gráfico que se muestra en la Fig.  11.1 . Abrir imagen en nueva ventanaFigura 11.1 Figura 11.1 Gráfico de la función y= 2 +1X2

Como ejemplo más complicado, considere una función que usaremos como parte de un ejercicio más amplio en la Secta. 11.6 . La siguiente es una función de utilidad esperada para un partido político en un modelo de cómo los partidos elegirán una posición de tema cuando compitan en elecciones secuenciales (Monogan 2013b, Eq. (6)): miUA(θA,θD)=Λ { - (metro1-θA)2+ (metro1-θD)2+ V}+ δΛ { - (metro2-θA)2+ (metro2-θD)2} (11,1) En esta ecuación, la utilidad esperada para un partido político (partido A ) es la suma de dos funciones de distribución logística acumuladas (las funciones Λ ), con la segunda ponderada por un término de descuento (0 ≤  δ  ≤ 1). La utilidad depende de las posiciones tomadas por el partido A y el partido D ( θ A y θ D ), una ventaja de valencia para el partido A ( V ) y la posición de emisión del votante mediano en la primera y segunda elección ( m 1 y m 2 ). Esta es ahora una función de varias variables y requiere que usemos el CDF de la distribución logística. Para ingresar esta función en R, escribimos: Función cuadrática.A <(m.1, m.2, p, delta, theta.A, theta.D) {

util.a <-plogis (- (m.1-theta.A) ^ 2 +  
  
 (m.1-theta.D) ^ 2 + p) +  
  
 delta \* plogis (- (m.2-theta.A) ^ 2 +  
  
 (m.2-theta.D) ^ 2)  
  
volver (util.a)  
  
}

El plogis comando calcula cada relevante p robabilidad del Logis tic CDF, y todos los demás términos se denominan auto-evidente de la ecuación. ( 11.1 ) (excepto que p se refiere al término de valencia V ). Aunque esta función era más complicada, todo lo que teníamos que hacer era asegurarnos de nombrar cada entrada y copiar Eq. ( 11.1 ) completamente en el cuerpo de la función.

Esta función más compleja, Quadratic.A , todavía se comporta como nuestra función simple. Por ejemplo, podríamos seguir adelante y proporcionar un valor numérico para cada argumento de la función como este:

Cuadrático.A (m.1 = .7, m.2 = -. 1, p = .1, delta = 0, theta.A = .7, theta.D = .7)

Al hacerlo, se imprime una salida de:

[1] 0.5249792

Por lo tanto, ahora sabemos que 0.52 es la utilidad esperada para la parte A cuando los términos toman estos valores numéricos. De forma aislada, este resultado no significa mucho. En teoría de juegos, normalmente estamos interesados ​​en cómo el partido A (o cualquier jugador) puede maximizar su utilidad. Por lo tanto, si tomamos todos los parámetros como fijos en los valores anteriores, excepto que permitimos que la parte A elija θ A como mejor le parezca, podemos visualizar cuál es la mejor opción de A. El siguiente código produce el gráfico que se ve en la figura  11.2 : Abrir imagen en nueva ventanaFigura 11.2 Figura 11.2 Gráfico de la utilidad esperada de un partido aventajado durante dos elecciones dependiendo de la posición del asunto

posiciones <-seq (-1,1, .01)

util.A <-Quadratic.A (m.1 = .7, m.2 = -. 1, p = .1, delta = 0,

theta.A = posiciones, theta.D = .7)

plot (x = posiciones, y = util.A, type = “l,”

xlab = "Posición del Partido A", ylab = "Utilidad")

En la primera línea, generamos una serie de posiciones del partido Un podrá optar, para θ A . En la segunda línea, calculamos un vector de utilidades esperadas para la parte A en cada una de las posiciones de emisión que consideramos y establecemos los otros parámetros en su valor mencionado anteriormente. Por último, usamos la función plot para dibujar un gráfico lineal de estas utilidades. Resulta que el máximo de esta función está en θ A  = 0. 7, por lo que nuestra utilidad de 0.52 mencionada anteriormente es la mejor fiesta que A puede hacer después de todo.

11.3 Bucles Los bucles son fáciles de escribir en R y se pueden usar para repetir cálculos que son idénticos o varían solo en unos pocos parámetros. La estructura básica de un bucle que utiliza el comando for es:

para (i en 1: M) {COMANDOS}

En este caso, M es el número de veces que se ejecutarán los comandos. R también admite bucles con el comando while que sigue una estructura de comando similar:

j <- 1

mientras que (j <M) {

COMANDOS  
  
 j <- j + 1  
  
 }

Si una de bucle o un tiempo de bucle funciona mejor puede variar según la situación, por lo que el usuario debe usar su propio juicio al elegir una configuración. En general, mientras que los bucles tienden a ser mejor para los problemas en los que se tendría que hacer algo hasta que se cumpla un criterio (como un criterio de convergencia), y para los bucles son generalmente mejores para las cosas que desea repetir un número fijo de veces. Bajo cualquier estructura, R permitirá que se incluya una amplia gama de comandos en un bucle; la tarea del usuario es cómo administrar la entrada y salida del bucle de manera eficiente.

Como demostración simple de cómo funcionan los bucles, considere un ejemplo que ilustra la ley de los grandes números. Para hacer esto, podemos simular observaciones r andom a partir del comando de distribución normal estándar (fácil con el comando rnorm ). Dado que la normal estándar tiene una media poblacional de cero, esperaríamos que la media muestral de nuestros valores simulados sea cercana a cero. A medida que el tamaño de nuestra muestra aumenta, la media muestral debería estar más cerca de la media poblacional de cero. Un bucle es perfecto para este ejercicio: queremos repetir los cálculos de la simulación a partir de la distribución normal estándar y luego tomar la media de las simulaciones. Lo que difiere de una iteración a otra es que queremos que el tamaño de nuestra muestra aumente.

Podemos configurar esto fácilmente con el siguiente código:

set.seed (271828183)

almacenar <- matriz (NA, 1000,1)

para (i en 1: 1000) {

a <- norma (i)  
  
 almacenar [i] <- media (a)  
  
 }

plot (store, type = “h,” ylab = “Sample Mean,”

xlab = "Número de observaciones")

abline (h = 0, col = ‘rojo,’ lwd = 2)

En la primera línea de este código, llamamos al comando set.seed , para que nuestro experimento de simulación sea replicable. Cuando R extrae números aleatoriamente de alguna manera, utiliza un generador de números pseudoaleatorios, que es una lista de 2,1 mil millones de números que se asemejan a extracciones aleatorios. 3 Al elegir cualquier número del 1 al 2,147,483,647, otros deberían poder reproducir nuestros resultados usando los mismos números en su simulación. Nuestra elección de 271,828,183 fue en gran medida arbitraria. En la segunda línea de código, creamos un vector en blanco llamado store de longitud 1000. Este vector es donde almacenaremos nuestra salida del ciclo. En las siguientes cuatro líneas de código, definimos nuestro ciclo. El ciclo va de 1 a 1000, y el índice de cada iteración se denominayo . En cada iteración, nuestro tamaño de muestra es simplemente el valor de i , por lo que la primera iteración simula una observación y la milésima iteración simula 1000 observaciones. Por tanto, el tamaño de la muestra aumenta con cada pasada del bucle. En cada pasada del programa, R toma muestras de un norte( 0 , 1 ) distribución y luego toma la media de esa muestra. Cada media se registra en la i- ésima celda de almacenamiento . Una vez que se cierra el ciclo, graficamos nuestras medias muestrales contra el tamaño de la muestra y usamos abline para dibujar una línea roja en la media poblacional de cero. El resultado se muestra en la figura  11.3 . De hecho, nuestro gráfico muestra la media de la muestra que converge a la media real de cero a medida que aumenta el tamaño de la muestra. Abrir imagen en nueva ventanaFigura 11.3 Figura 11.3 La ley de los grandes números y los bucles en acción.

Los bucles son necesarios para muchos tipos de programas, por ejemplo, si desea hacer un análisis de Monte Carlo. En algunos casos (pero no en todos), sin embargo, los bucles pueden ser más lentos que las versiones vectorizadas de los comandos. Puede valer la pena probar el comando apply , por ejemplo, si puede lograr el mismo objetivo que un bucle. Cuál es más rápido a menudo depende de la complejidad de la función y la sobrecarga de memoria para calcular y almacenar resultados. Entonces, si un enfoque es demasiado lento o demasiado complicado para que su computadora lo maneje, puede valer la pena probar el otro.

11.4 Ramificación Los usuarios de R tienen la opción de ejecutar comandos condicionados a una declaración booleana usando el comando if . Esto puede ser útil siempre que el usuario solo quiera implementar algo para ciertos casos, o si los diferentes tipos de datos deben tratarse de manera diferente. La sintaxis básica de una instrucción if es:

si (expresión\_lógica) {

expresión\_1  
  
 ...

}

En este marco, la expresión\_lógica es una declaración booleana, y la expresión\_1 representa los comandos que el usuario quisiera aplicar siempre que la expresión lógica sea verdadera.

Como un ejemplo de juguete de cómo funciona esto, supongamos que quisiéramos simular un proceso en el que dibujamos dos números del 1 al 10 (con números repetidos permitidos). El comando de muestra nos permite simular este proceso fácilmente, y si lo incrustamos en un bucle for podemos repetir el experimento varias veces (digamos, 100 intentos). Si quisiéramos saber en cuántos ensayos aparecieron ambos números, podríamos determinar esto con una declaración if . Nuestro código se junta así:

even.count <-0

para (i en 1: 100) {

a <-muestra (c (1:10), 2, reemplazar = VERDADERO)  
  
 si (suma (a %% 2) == 0) {  
  
 even.count <-even.count + 1  
  
 }

}

even.count

La primera línea crea un escalar llamado even.count y lo establece en cero. La siguiente línea inicia el ciclo que nos da 100 pruebas. La tercera línea crea nuestra muestra de dos números del 1 al 10 y nombra la muestra a . La cuarta línea define nuestra declaración if : usamos la función módulo para encontrar el resto cuando cada término de nuestra muestra se divide por dos. 4 Si el resto de cada término es cero, entonces todos los números son pares y la suma es cero. En ese caso, hemos extraído una muestra en la que los números son pares. Por lo tanto, cuando sum (a %% 2) == 0es cierto, entonces queremos agregar uno a un recuento continuo de cuántas muestras de dos números pares tenemos. Por lo tanto, la quinta línea de nuestro código se suma a nuestro recuento, pero solo en los casos que cumplen con nuestra condición. (Observe que una asignación recursiva para even.count es aceptable. R tomará el valor anterior, le agregará uno y actualizará el valor guardado). Pruebe este experimento usted mismo. Como sugerencia, la teoría de la probabilidad diría que el 25% de los ensayos producirán dos números pares, en promedio.

Los usuarios también pueden hacer declaraciones if … else de bifurcación de modo que un conjunto de operaciones se aplique siempre que una expresión sea verdadera, y otro conjunto de operaciones se aplique cuando la expresión sea falsa. Esta estructura básica se ve así:

si (expresión\_lógica) {

expresión\_1  
  
 ...

} demás {

expresión\_2  
  
 ...

}

En este caso, expression\_1 solo se aplicará en los casos en que la expresión lógica sea verdadera, y expression\_2 solo se aplicará en los casos en que la expresión lógica sea falsa.

Finalmente, los usuarios tienen la opción de ramificarse aún más. En los casos en los que la primera expresión lógica es falsa, por lo que se llama a las expresiones que siguen a else , estos casos se pueden ramificar de nuevo con otra instrucción if … else . De hecho, el programador puede anidar tantas declaraciones if … else como desee. Para ilustrar esto, considere nuevamente el caso cuando simulamos dos números aleatorios del 1 al 10. Imagine que esta vez queremos saber no solo con qué frecuencia sacamos dos números pares, sino también con qué frecuencia sacamos dos números impares y con qué frecuencia sacamos un número par y uno impar. Una forma en que podríamos abordar esto es manteniendo tres recuentos en ejecución (a continuación, denominados even.count ,odd.count y split.count ) y agregando declaraciones de bifurcación adicionales:

even.count <-0

cuenta impar <-0

split.count <-0

para (i en 1: 100) {

a <-muestra (c (1:10), 2, reemplazar = VERDADERO)  
  
 si (suma (a %% 2) == 0) {  
  
 even.count <-even.count + 1  
  
 } más si (suma (a %% 2) == 2) {  
  
 recuento.imor <-cuenta.odd + 1  
  
 } demás{  
  
 split.count <-split.count + 1  
  
 }

}

even.count

cuenta impar

split.count

Nuestro ciclo for comienza igual que antes, pero después de nuestra primera instrucción if , seguimos con una instrucción else . Cualquier muestra que no consista en dos números pares ahora está sujeta a los comandos debajo de else , y el primer comando debajo de else es … otra instrucción if . El siguiente enunciado if observa que si ambos términos de la muestra son impares, la suma de los residuos después de dividir por dos será dos. Por lo tanto, todas las muestras con dos entradas impares ahora están sujetas a los comandos de esta nueva instrucción if , donde vemos que nuestro índice de cuenta impar aumentará en uno. Por último, tenemos un resto finalinstrucción: todas las muestras que no constan de dos números pares o impares ahora estarán sujetas a este conjunto final de comandos. Dado que estas muestras constan de un número par y uno impar, el índice split.count aumentará en uno. Pruébelo usted mismo. Nuevamente, como sugerencia, la teoría de la probabilidad indica que, en promedio, el 50% de las muestras debe constar de un número par y uno impar, el 25% debe constar de dos números pares y el 25% debe constar de dos números impares.

11.5 Optimización y estimación de máxima verosimilitud El comando optim en R permite a los usuarios encontrar el mínimo o máximo de una función usando una variedad de métodos de optimización numérica . 5 En otras palabras, optim tiene varios algoritmos que buscan de manera eficiente el valor más alto o más bajo de una función, varios de los cuales permiten al usuario restringir los valores posibles de las variables de entrada. Si bien hay muchos usos de la optimización en la investigación de ciencias políticas, el más común es la estimación de máxima verosimilitud (MLE). 6

El comando optim , junto con la definición de función simple de R (discutida en la Sección  11.2 ), le permite a R usar fácilmente la flexibilidad total de MLE. Si un modelo enlatado como los descritos en los Cap. 1 –7 no se adapta a su investigación y desea derivar su propio estimador utilizando MLE, entonces R puede adaptarse a usted.

Para ilustrar cómo funciona la programación de un MLE en R , considere un ejemplo simple: el estimador del parámetro de probabilidad ( π ) en una distribución binomial. A modo de recordatorio, la motivación detrás de una distribución binomial es que estamos interesados ​​en una prueba que toma uno de dos resultados cada vez, y repetimos esa prueba varias veces. El ejemplo más simple es que lanzamos una moneda, que saldrá cara o cruz en cada lanzamiento. Usando este ejemplo, suponga que nuestros datos consisten en 100 lanzamientos de moneda y 43 de los lanzamientos salieron cara. ¿Cuál es la estimación MLE de la probabilidad de que salga cara? Por intuición o derivación, debes saber que π^=43100= 0,43 . Sin embargo, seguiremos estimando esta probabilidad en R para practicar en casos más complicados.

Para definir más formalmente nuestro problema, una distribución binomial se motiva al completar n ensayos de Bernoulli, o ensayos que pueden terminar en un éxito o un fracaso. Registramos el número de veces que tenemos éxito en nuestras n pruebas como y . Además, por definición, nuestro parámetro de probabilidad ( π ) debe ser mayor o igual a cero y menor o igual a uno. Nuestra función de verosimilitud es: L ( π| n , y) =πy( 1 - π)n - y (11,2) Para facilitar el cálculo y el cálculo, podemos obtener un resultado equivalente maximizando nuestra función logarítmica de verosimilitud: ℓ ( π| n , y) = registroL ( π| n , y) = y⋅ registro( π) + ( n - y) registro( 1 - π) (11,3) Podemos definir fácilmente nuestra función de probabilidad logarítmica en R con el comando de función : binomial.loglikelihood <- función (prob, y, n) {

loglikelihood <- y \* log (prob) + (ny) \* log (1-prob)

retorno (loglikelihood)

}

Ahora para estimar π^ , necesitamos R para encontrar el valor de π que maximiza la función logarítmica de verosimilitud dados los datos. (Llamamos a este término prob en el código para evitar confusiones con el uso que hace R del comando pi para almacenar la constante geométrica). Podemos hacer esto con el comando optim . Calculamos:

test <- optim (c (.5), # valor inicial para prob

binomial.loglikelihood, # la función logarítmica de verosimilitud

method = “BFGS,” # método de optimización

arpillera = VERDADERO, # devuelve arpillera numérica

control = lista (fnscale = -1), # maximizar en lugar de minimizar

y = 43, n = 100) # los datos

imprimir (prueba)

Recuerda que todo lo que sigue a un signo de almohadilla ( # ) es un comentario que R ignora, por lo que estas notas sirven para describir cada línea de código. Siempre comenzamos con un vector de valores iniciales con todos los parámetros para estimar (solo π en este caso), nombramos la función de probabilidad logarítmica que definimos en otro lugar, elegimos nuestro método de optimización ( B royden- F letcher- G oldfarb- S hanno es a menudo una buena elección) e indicar que queremos que R devuelva el hessiano numérico para poder calcular los errores estándar más tarde. La quinta línea de código es de vital importancia: de forma predeterminada, optim es un minimizador , por lo que tenemos que especificarfnscale = -1 para convertirlo en un maximizador . Siempre que utilice optim para la estimación de máxima verosimilitud, deberá incluir esta línea. En la sexta línea, enumeramos nuestros datos. A menudo, aquí llamaremos una matriz o un marco de datos, pero en este caso solo necesitamos enumerar los valores de y y n .

Nuestro resultado de impresión (prueba) se ve así:

$ par

[1] 0,4300015

$ valor

[1] -68.33149

$ cuenta

gradiente de función

13 4

$ convergencia

[1] 0

$ mensaje

NULO

$ arpillera

[, 1]

[1,] -407,9996

Para interpretar nuestra salida: El término par enumera las estimaciones de los parámetros, por lo que nuestra estimación es π^= 0,43 (como anticipamos). La probabilidad logarítmica de nuestra solución final es - 68. 33149, y se presenta bajo valor . El término cuenta nos dice con qué frecuencia optim tuvo que llamar a la función y al gradiente. El término convergencia se codificará como 0 si la optimización se completó con éxito; cualquier otro valor es un código de error. El elemento de mensaje puede devolver otra información necesaria del optimizador. Por último, la arpillera es nuestra matriz de segundas derivadas de la función logarítmica de verosimilitud. Con solo un parámetro aquí, es una matriz simple de 1 × 1. En general, si el usuario quiere errores estándar a partir de un modelo de máxima verosimilitud estimada, la siguiente línea los devolverá:

sqrt (diag (resolver (-prueba $ arpillera)))

Esta línea se basa en la fórmula para errores estándar en la estimación de máxima verosimilitud. Todo lo que el usuario necesitará cambiar es reemplazar la palabra prueba con el nombre asociado con la llamada a optim . En este caso, R informa que el error estándar es S E (π^) = 0.0495074 .

Finalmente, en este caso en el que tenemos un solo parámetro de interés, tenemos la opción de usar nuestra función de probabilidad logarítmica definida para dibujar una imagen del problema de optimización. Considere el siguiente código:

regla <- seq (0,1,0.01)

loglikelihood <- binomial.loglikelihood (regla, y = 43, n = 100)

plot (regla, loglikelihood, type = “l,” lwd = 2, col = “blue,”

xlab = expresión (pi), ylab = “Log-Likelihood,” ylim = c (-300, -70),

main = “Log-Likelihood for Binomial Model”)

abline (v = .43)

La primera línea define todos los valores que posiblemente pueda tomar π . La segunda línea inserta en la función logarítmica de verosimilitud el vector de valores posibles de π , más los valores verdaderos de y y n . Las líneas tercera a quinta son una llamada a graficar que nos da un gráfico lineal de la función logarítmica de verosimilitud. La última línea dibuja una línea vertical en nuestro valor estimado para π^ . El resultado de esto se muestra en la Fig.  11.4 . Si bien las funciones log-verosimilitud con muchos parámetros generalmente no se pueden visualizar, esto nos recuerda que la función que hemos definido aún se puede usar para fines distintos a la optimización, si es necesario. Abrir imagen en nueva ventanaFigura 11.4 Figura 11.4 Logaritmo de verosimilitud binomial en todos los valores posibles del parámetro de probabilidad π cuando los datos constan de 43 éxitos en 100 ensayos

11.6 Programación orientada a objetos Esta sección da un giro hacia lo avanzado, al presentar una aplicación en programación orientada a objetos. Si bien esta aplicación sintetiza muchas de las otras herramientas discutidas en este capítulo, es menos probable que los usuarios novatos utilicen la programación orientada a objetos que las características discutidas hasta ahora. Dicho esto, para usos avanzados, el trabajo orientado a objetos puede ser beneficioso.

Como se mencionó en el Cap. 1 , R es un entorno orientado a objetos. Esto significa que usted, como investigador, tiene la oportunidad de utilizar sofisticadas herramientas de programación en su propia investigación. En la programación orientada a objetos, crea una clase de elemento que tiene una variedad de características. Luego, puede crear elementos dentro de la clase (llamados objetos o variables ) que tendrán características únicas. En R , puede crear clases a partir de los sistemas de objetos S3 o S4 .

Para ilustrar, un “modelo lineal” es una clase de objetos creados por el comando lm . Es una clase S3 . De vuelta en el Cap. 6 , estimamos un modelo que llamamos mod.hours , que era un objeto de la clase de modelo lineal. Este objeto tenía características que incluyen coeficientes , residuos y cov . Sin escala . En todos los objetos de la clase S3 , podemos llamar a una característica escribiendo el nombre del objeto, el signo de dólar y el nombre de la característica que queremos. Por ejemplo, mod.hours $ coefficients enumeraría el vector de coeficientes de ese modelo. ( T4 utiliza referencias ligeramente diferentes, que se describen más adelante.) Cuando defina sus propios objetos, puede utilizar el sistema de objetos S3 o S4 . El siguiente ejemplo guarda nuestras salidas en el sistema de objetos S3 , aunque las notas al pie ilustran cómo podría usar el sistema de objetos S4 si lo prefiere.

11.6.1 Simular un juego Como ejemplo práctico de programación orientada a objetos, consideramos una versión algo simplificada del trabajo presentado en Monogan (2013b). 7 Tocamos este modelo en la Secta. 11.2 , presentando la función de utilidad de un actor. En resumen, este artículo desarrolla un modelo de teoría de juegos de cómo dos partes elegirán una posición sobre un tema de acuerdo con el modelo de proximidad espacial de la política. El modelo espacial, descrito algo en el Cap. 7 , asume que las posiciones de los problemas se pueden representar como ubicaciones en el espacio. Los votantes eligen partidos que adoptan posiciones de emisión más cercanas a su propia posición en el espacio de emisión, por lo que los partidos tratan de maximizar los votos eligiendo estratégicamente sus posiciones de emisión.

En el juego que consideramos, la motivación sustantiva es el hecho de que la opinión pública muestra fuertes tendencias en ciertos temas de política debido a tendencias demográficas o diferencias generacionales. Temas como este incluyen la política ambiental, la política de inmigración y la protección de los derechos de los homosexuales. Suponemos que dos partidos competirán en dos elecciones. Sin embargo, para considerar el hecho de que los partidos deben tener en cuenta el futuro y pueden rendir cuentas por posiciones de asuntos pasados, la posición que toman en la primera elección es la posición en la que están atrapados en la segunda elección. También asumimos que la posición del votante mediano cambia de una elección a la siguiente de una manera que los partidos anticipan, ya que en estos temas las tendencias de la opinión pública suelen ser claras.

Otra característica es que un partido tiene ventaja en la primera elección (debido a la titularidad o algún otro factor), por lo que los jugadores son el partido A (un partido que no tiene ventaja de valencia en la primera elección) y el partido D (una parte desfavorecida). Se supone que la segunda elección es menos valiosa que la primera, porque la recompensa está más lejos. Finalmente, los votantes también consideran factores desconocidos para los partidos, por lo que el resultado de la elección es más probabilístico que seguro. En este juego, entonces, los partidos intentan maximizar su probabilidad de ganar cada una de las dos elecciones. ¿Cómo se posicionan ante un tema que cambia la opinión pública?

La característica engañosa de este juego es que no se puede resolver algebraicamente (Monogan 2013b, pag. 288). Por lo tanto, recurrimos a R para ayudarnos a encontrar soluciones a través de la simulación. Lo primero que debemos hacer es limpiar y definir las funciones de utilidad esperadas para las partes A y D :

rm (lista = ls ())

Función cuadrática.A <(m.1, m.2, p, delta, theta.A, theta.D) {

util.a <-plogis (- (m.1-theta.A) ^ 2 +  
  
 (m.1-theta.D) ^ 2 + p) +  
  
 delta \* plogis (- (m.2-theta.A) ^ 2 +  
  
 (m.2-theta.D) ^ 2)  
  
volver (util.a)  
  
}

D <-función cuadrática (m.1, m.2, p, delta, theta.A, theta.D) {

util.d <- (1-plogis (- (m.1-theta.A) ^ 2 +  
  
 (m.1-theta.D) ^ 2 + p)) +  
  
 delta \* (1-plogis (- (m.2-theta.A) ^ 2 +  
  
 (m.2-theta.D) ^ 2))  
  
volver (util.d)  
  
}

Ya consideramos la función Quadratic.A en la secc. 11.2 , y está formalmente definido por la Ec. ( 11,1 ). Quadratic.D es similar en estructura, pero produce diferentes utilidades. Ambas funciones se utilizarán en las simulaciones que hagamos.

Nuestro siguiente paso es definir una función larga que usa nuestras funciones de utilidad para ejecutar una simulación y produce una salida con el formato que queremos. 8 Todo el siguiente código es una gran definición de función. Los comentarios después de los signos de almohadilla ( # ) se incluyen nuevamente para ayudar a distinguir cada parte del cuerpo de la función. La función, llamada simular , simula nuestro juego de interés y guarda nuestros resultados en un objeto de clase game.simulation :

simular <-función (v, delta, m.2, m.1 = 0.7, theta = seq (-1,1, .1)) {

#definir parámetros internos, matrices y vectores  
  
precisión <-longitud (theta)  
  
resultA <-matrix (NA, precisión, precisión)  
  
resultD <-matrix (NA, precisión, precisión)  
  
bestResponseA <-rep (NA, precisión)  
  
bestResponseD <-rep (NA, precisión)  
  
equilibrioA <- 'NA'  
  
equilibrioD <- 'NA'  
  
#matrix atributos  
  
nombres de fila (resultA) <- colnames (resultA) <- nombres de fila (resultD) <-  
  
 colnames (resultD) <- nombres (bestResponseA) <-  
  
 nombres (bestResponseD) <- theta  
  
# Complete las utilidades para todas las estrategias para el partido A  
  
para (i en 1: precisión) {  
  
 para (j en 1: precisión) {  
  
 resultadoA [i, j] <- Cuadrático.A (m.1, m.2, v, delta,  
  
 theta [i], theta [j])  
  
 }  
  
 }  
  
#utilidades para la fiesta D  
  
para (i en 1: precisión) {  
  
 para (j en 1: precisión) {  
  
 resultadoD [i, j] <- Cuadrático.D (m.1, m.2, v, delta,  
  
 theta [i], theta [j])  
  
 }  
  
 }  
  
#mejores respuestas para la fiesta A  
  
para (i en 1: precisión) {  
  
 bestResponseA [i] <- which.max (resultA [, i])  
  
 }  
  
#mejores respuestas para la fiesta D  
  
para (i en 1: precisión) {  
  
 bestResponseD [i] <- which.max (resultD [i,])  
  
 }  
  
#encuentra los equilibrios  
  
para (i en 1: precisión) {  
  
 if (bestResponseD [bestResponseA [i]] == i) {  
  
 equilibriumA <-dimnames (resultA) [[1]] [  
  
 bestResponseA [i]]  
  
 equilibriumD <-dimnames (resultD) [[2]] [  
  
 bestResponseD [bestResponseA [i]]]  
  
 }  
  
}  
  
# guardar la salida  
  
resultado <-lista (resultadoA = resultadoA, resultadoD = resultadoD,  
  
 bestResponseA = bestResponseA, bestResponseD = bestResponseD,  
  
 equilibrio A = equilibrio A, equilibrio D = equilibrio D)  
  
clase (resultado) <- "juego.simulación"  
  
invisible (resultado)

}

Como consejo general, al escribir una función larga, es mejor probar el código fuera del contenedor de la función . Para obtener puntos de bonificación y tener una idea completa de este código, es posible que desee dividir esta función en sus componentes y ver cómo funciona cada pieza. Como muestran los comentarios a lo largo de todo, cada conjunto de comandos hace algo que normalmente haríamos en una función. Observe que cuando se definen los argumentos, tanto m.1 como thetase les dan valores predeterminados. Esto significa que en usos futuros, si no especificamos los valores de estas entradas, la función utilizará estos valores predeterminados, pero si los especificamos, los valores predeterminados se anularán. Pasando a los argumentos dentro de la función, comienza definiendo parámetros internos y estableciendo sus atributos. Cada conjunto de comandos a partir de entonces es un ciclo dentro de un ciclo para repetir ciertos comandos y llenar vectores y matrices de salida.

La adición clave aquí que es exclusiva de todo lo que se discutió anteriormente está en la definición del término de resultado al final de la función. Observe que al hacer esto, primero definimos la salida como una lista , en la que se nombra cada componente. En este caso, ya nombramos nuestros objetos de la misma manera que queremos etiquetarlos en la lista de salida, pero puede que no siempre sea así. 9 Luego usamos el comando class para declarar que nuestra salida es de la clase game.simulation , un concepto que estamos creando ahora. El comando de clase lo formatea como un objeto de la familia S3 . Al escribir invisible (resultado)al final de la función, sabemos que nuestra función devolverá este objeto game.simulation -class . 10

Ahora que esta función está definida, podemos ponerla en práctica. Refiriéndose a los términos de la Ec. ( 11.1 ), suponga que V  = 0, 1, δ  = 0, m 1  = 0, 7 y m 2  = −0. 1. En el siguiente código, creamos un nuevo objeto llamado tratamiento.1 usando la función de simulación cuando los parámetros toman estos valores, y luego le pedimos a R que imprima la salida:

tratamiento.1 <-simular (v = 0.1, delta = 0.0, m.2 = -0.1)

tratamiento.1

Observe que no especificamos m.1 porque 0.7 ya es el valor predeterminado para ese parámetro. El resultado de la impresión de treatment.1 es demasiado extenso para reproducirlo aquí, pero en su propia pantalla verá que treatment.1 es un objeto de la clase game.simulation , y la impresión informa los valores de cada atributo.

Si solo estamos interesados ​​en una característica particular de este resultado, podemos pedirle a R que devuelva solo el valor de una ranura específica. Para un objeto S3 , que es cómo guardamos nuestro resultado, podemos llamar a un atributo nombrando el objeto, usando el símbolo $ y luego nombrando el atributo que queremos usar. (Esto contrasta con los objetos S4 , que usan el símbolo @ para llamar ranuras). Por ejemplo, si solo quisiéramos ver cuáles eran las opciones de equilibrio para las partes A y D , simplemente podríamos escribir:

tratamiento.1 $ equilibrioA

tratamiento.1 $ equilibrioD

Cada uno de estos comandos devuelve el mismo resultado:

[1] “0,7”

Entonces, sustancialmente, podemos concluir que el equilibrio para el juego bajo estos parámetros es θ A  =  θ D  = 0. 7, que es el punto ideal del votante mediano en la primera elección. En esencia, esto debería tener sentido porque establecer δ  = 0 es un caso especial cuando los partidos no están en absoluto preocupados por ganar la segunda elección, por lo que se posicionan estrictamente para la primera elección. 11

Para dibujar un contraste, ¿qué pasa si realizamos una segunda simulación, pero esta vez aumentamos el valor de δ a 0.1? También podríamos usar valores más detallados de las posiciones que podrían tomar las partes, estableciendo su posición hasta las centésimas de decimal, en lugar de décimas. Podríamos hacer esto de la siguiente manera:

tratamiento.2 <-simular (v = 0.1, delta = 0.1, m.2 = -0.1,

theta = seq (-1,1, .01))

tratamiento.2 $ equilibrioA

tratamiento.2 $ equilibrioD

Logramos la precisión más fina al sustituir nuestro propio vector por theta . Nuestros valores de salida de las dos ranuras de equilibrio son nuevamente los mismos:

[1] “0,63”

Entonces sabemos que bajo este segundo tratamiento, el equilibrio para el juego es θ A  =  θ D  = 0. 63. Sustancialmente, lo que sucedió aquí es que aumentamos el valor de ganar la segunda elección para los partidos. Como resultado, los partidos movieron su posición de tema un poco más cerca de la preferencia de tema del votante medio en la segunda elección.

11.7 Análisis de Monte Carlo: un ejemplo aplicado Como ejemplo aplicado que sintetiza varias de las herramientas desarrolladas en este capítulo, ahora realizamos un análisis de Monte Carlo. La lógica básica del análisis de Monte Carlo es que el investigador genera datos conociendo el verdadero modelo de población. Luego, el investigador pregunta si las estimaciones muestrales de un determinado método tienen buenas propiedades, dadas las cantidades de población. Los tratamientos comunes en un experimento de Monte Carlo incluyen: elección de estimador, tamaño de muestra, distribución de predictores, varianza de errores y si el modelo tiene la forma funcional correcta (por ejemplo, ignorar una relación no lineal u omitir un predictor). Al probar varios tratamientos, un usuario puede tener una idea comparativa de qué tan bien funciona un estimador.

En este caso, presentaremos Signorino’s (1999) modelo probit multinomial estratégico como estimador que usaremos en nuestro experimento de Monte Carlo. La idea detrás de este enfoque es que, cuando una situación política recurrente se puede representar con un juego (como las disputas internacionales militarizadas), podemos desarrollar un modelo empírico de los resultados que pueden ocurrir (como la guerra) basado en la estrategia del gobierno. juego. Cada resultado posible tiene una función de utilidadpara cada jugador, ya sea que el jugador sea un individuo, un país u otro actor. La utilidad representa cuánto beneficio obtiene el jugador del resultado. En esta configuración, utilizamos predictores observables para modelar las utilidades para cada resultado posible. Este tipo de modelo es interesante porque el proceso de generación de datos de población generalmente tiene no linealidades que no son capturadas por enfoques estándar. Por lo tanto, tendremos que programar nuestra propia función de probabilidad y usar optim para estimarla. 12

Para motivar cómo se establece el probit estratégico multinomial, considere un modelo sustantivo de comportamiento entre dos naciones, un agresor (1) y un objetivo (2). Este es un juego de dos etapas: primero, el agresor decide si atacar (A) o no (∼ A); entonces, el objetivo decide si defender (D) o no (∼ D). Pueden ocurrir tres consecuencias observables: si el agresor decide no atacar, el status quo se mantiene; si el agresor ataca, el objetivo puede optar por capitular ; finalmente, si el objetivo defiende, tenemos guerra . El árbol del juego se presenta en la Fig.  11.5 . Al final de cada rama están las utilidades para los jugadores (1) y (2) para el status quo (SQ), la capitulación (C) y la guerra.(W), respectivamente. Suponemos que estos jugadores son racionales y, por lo tanto, elegimos las acciones que maximizan la recompensa resultante. Por ejemplo, si el objetivo es débil, la recompensa de la guerra U 2 ( W ) sería terriblemente negativa y probablemente menor que la recompensa de la capitulación U 2 ( C ). Si el agresor lo sabe, puede deducir que en caso de ataque, el objetivo capitularía. Por lo tanto, el agresor sabría que si elige atacar, recibirá la recompensa U 1 ( C ) (y no U 1 ( W )). Si la recompensa de la capitulación U 1 (C ) es más grande que la recompensa del status quo U 1 (SQ) para el agresor, la decisión racional es atacar (y la decisión racional del objetivo es capitular). Por supuesto, el objetivo es tener una idea de lo que sucederá a medida que cambien las circunstancias en diferentes díadas de naciones. Abrir imagen en nueva ventanaFigura 11.5 Figura 11.5 Modelo de disuasión estratégica

En general, asumimos que las funciones de utilidad responden a circunstancias específicas observables (el valor de los recursos en disputa, el precio diplomático esperado debido a las sanciones en caso de agresión, el poderío militar, etc.). También asumiremos que las utilidades para la elección de cada país son estocásticas. Esto introduce incertidumbre en nuestro modelo de pagos, lo que representa el hecho de que este modelo está incompleto y probablemente excluye algunos parámetros relevantes de la evaluación. En este ejercicio, consideraremos solo cuatro predictores X i para modelar las funciones de utilidad y asumiremos que los pagos son lineales en estas variables.

En particular, consideramos la siguiente forma paramétrica para los pagos: U1( S Q )U1( C)U1( W)U2( C)U2( W)α=====∼0X1β1X2β20X3β3+X4β4norte( 0 , 0,5 ) (11,4) Cada nación hace su elección basándose en qué decisión le dará una mayor utilidad. Esto se basa en la información conocida, más una perturbación privada desconocida ( α ) para cada elección. Esta perturbación privada agrega un elemento aleatorio a las decisiones de los actores. Nosotros, como investigadores, podemos mirar los conflictos pasados, medir los predictores ( X ), observar el resultado y nos gustaría inferir: ¿Qué importancia tienen X 1 , X 2 , X 3 y X 4 en el comportamiento de las díadas-nación? Tenemos que determinar esto basándonos en si cada punto de datos resultó en un status quo , una capitulación o una guerra . 11.7.1 Función log-verosimilitud de disuasión estratégica Podemos determinar estimaciones de cada β^ utilizando la estimación de máxima verosimilitud. Primero, tenemos que determinar la probabilidad ( p ) que el objetivo (2) defenderá si es atacado: p = P( D )===PAG(U2( D ) +α2 D>U2( ∼ D ) +α2 ∼ D)PAG(U2( D ) -U2( ∼ D ) >α2 ∼ D-α2 D)Φ (X3β3+X4β4) (11,5) Donde Φ es la función de distribución acumulativa para una distribución normal estándar. Si conocemos p , podemos determinar la probabilidad ( q ) de que el agresor (1) ataque: q= P( A )===PAG(U1( A ) +α1 A>Ua( ∼ A ) +α1 ∼ A)PAG(U1( A ) -U1( ∼ A ) >α1 ∼ A-α1 A)Φ ( pX2β2+ ( 1 - p )X1β1) (11,6) Observe que p está en la ecuación de q . Esta característica no lineal del modelo no se adapta a los modelos enlatados estándar. Conocer las fórmulas para p y q , sabemos que las probabilidades de status quo, la capitulación, y la guerra. Por lo tanto, la función de verosimilitud es simplemente el producto de las probabilidades de cada evento, elevado a una variable ficticia de si el evento sucedió, multiplicado por todas las observaciones: L ( β | y , X ) =∏i = 1norte( 1 - q)DS Q× ( q( 1 - p ))DC× ( p q)DW (11,7) Donde D SQ , D C y D W son variables ficticias iguales a 1 si el caso es status quo, capitulación o guerra, respectivamente, y 0 en caso contrario. La función logarítmica de verosimilitud es: ℓ ( β | y , X ) =∑i = 1norteDS QIniciar sesión( 1 - q) +DCIniciar sesiónq( 1 - p ) +DWIniciar sesión( p q) (11,8) Ahora estamos listos para empezar a programar esto en R . Comenzamos limpiando y luego definiendo nuestra función log-verosimilitud como llik , que nuevamente incluye comentarios para describir partes del cuerpo de la función:

rm (lista = ls ())

llik = función (B, X, Y) {

#Separe matrices de datos para variables individuales:

sq = as.matrix (Y [, 1])

cap = as.matrix (Y [, 2])

war = as.matrix (Y [, 3])

X13 = as.matrix (X [, 1])

X14 = as.matrix (X [, 2])

X24 = as.matrix (X [, 3: 4])

# Coeficientes separados para cada ecuación:

B13 = como.matriz (B [1])

B14 = como.matriz (B [2])

B24 = como.matriz (B [3: 4])

# Definir utilidades como variables multiplicadas por coeficientes:

U13 = X13% \*% B13

U14 = X14% \*% B14

U24 = X24% \*% B24

#Calcular la probabilidad de que 2 pelee (P4) o no (P3):

P4 = pnorm (U24)

P3 = 1-P4

#Calcular la probabilidad de que 1 ataque (P2) o no (P1):

P2 = norma ((P3 \* U13 + P4 \* U14))

P1 = 1-P2

# Definir y devolver la función de probabilidad logarítmica:

lnpsq = log (P1)

lnpcap = log (P2 \* P3)

lnpwar = log (P2 \* P4)

llik = sq \* lnpsq + cap \* lnpcap + war \* lnpwar

retorno (suma (llik))

}

Aunque sustancialmente más largo que la función de verosimilitud que definimos en la Sec. 11.5 , la idea es la misma. La función aún acepta valores de parámetros, variables independientes y variables dependientes, y aún devuelve un valor de probabilidad logarítmica. Sin embargo, con un modelo más complejo, la función debe dividirse en pasos de componentes. Primero, nuestros datos ahora son matrices, por lo que el primer lote de código separa las variables por ecuación del modelo. En segundo lugar, nuestros parámetros son todos coeficientes almacenados en el argumento B, por lo que necesitamos separar los coeficientes por ecuación del modelo. En tercer lugar, multiplicamos por matrices las variables por los coeficientes para crear los tres términos de utilidad. Cuarto, usamos esa información para calcular las probabilidades de que el objetivo se defienda o no. En quinto lugar, utilizamos las utilidades, más la probabilidad de las acciones del objetivo, para determinar la probabilidad de que el agresor ataque o no. Por último, las probabilidades de todos los resultados se utilizan para crear la función de probabilidad logarítmica.

11.7.2 Evaluación del estimador Ahora que hemos definido nuestra función de verosimilitud, podemos usarla para simular datos y ajustar el modelo sobre nuestros datos simulados. Con cualquier experimento de Monte Carlo, debemos comenzar por definir el número de experimentos que realizaremos para un tratamiento determinado y el número de puntos de datos simulados en cada experimento. También necesitamos definir espacios vacíos para los resultados de cada experimento. Escribimos:

set.seed (3141593)

i <-100 # número de experimentos

n <-1000 # número de casos por experimento

beta.qre <-matriz (NA, i, 4)

stder.qre <-matriz (NA, i, 4)

Comenzamos usando set.seed para hacer que nuestros resultados de Monte Carlo sean más replicables. Aquí dejamos que i sea ​​nuestro número de experimentos, que establecemos en 100. (Aunque normalmente podríamos preferir un número mayor). Usamos n como nuestro número de casos. Esto nos permite definir beta.qre y stder.qre como las matrices de salida para nuestras estimaciones de los coeficientes y errores estándar de nuestros modelos, respectivamente.

Con esto en su lugar, ahora podemos ejecutar un gran ciclo que simulará repetidamente un conjunto de datos, estimará nuestro modelo probit estratégico multinomial y luego registrará los resultados. El bucle es el siguiente:

para (j en 1: i) {

#Simular variables causales  
  
 x1 <-rnorm (n)  
  
 x2 <-rnorm (n)  
  
 x3 <-rnorm (n)  
  
 x4 <-rnorm (n)  
  
 #Crear utilidades y términos de error  
  
 u11 <-rnorm (n, sd = sqrt (.5))  
  
 u13 <-x1  
  
 u23 <-rnorm (n, sd = sqrt (.5))  
  
 u14 <-x2  
  
 u24 <-x3 + x4 + rnorm (n, sd = sqrt (.5))  
  
 pR <-pnorm (x3 + x4)  
  
 uA <- (pR \* u14) + ((1-pR) \* u13) + rnorm (n, sd = sqrt (.5))  
  
 #Crear variables dependientes  
  
 sq <-rep (0, n)  
  
 capit <-rep (0, n)  
  
 guerra <-rep (0, n)  
  
 sq [u11> = uA] <- 1  
  
 capit [u11 <uA & u23> = u24] <- 1  
  
 guerra [u11 <uA & u23 <u24] <- 1  
  
 Nsq <-abs (1 cuadrado)  
  
 #Matrices para entrada  
  
 stval <-rep (.1,4)  
  
 depvar <-cbind (sq, capit, war)  
  
 indvar <-cbind (x1, x2, x3, x4)  
  
 #Modelo de ajuste  
  
 strat.mle <-optim (stval, llik, hessian = TRUE, method = "BFGS",  
  
 control = lista (maxit = 2000, fnscale = -1, trace = 1),  
  
 X = indvar, Y = depvar)  
  
 #Guardar resultados  
  
 beta.qre [j,] <- estrat.mle $ par  
  
 stder.qre [j,] <- sqrt (diag (solve (-strat.mle $ hessian)))

}

En este modelo, establecemos β  = 1 para cada coeficiente en el modelo de población. De lo contrario, Eq. ( 11.4 ) define completamente nuestro modelo de población. En el ciclo, los primeros tres lotes de código generan datos de acuerdo con este modelo. Primero, generamos cuatro variables independientes, cada una con una distribución normal estándar. En segundo lugar, definimos las utilidades de acuerdo con la ecuación. ( 11.4 ), agregando los términos de perturbación aleatoria ( α ) como se indica en la figura  11.5 . En tercer lugar, creamos los valores de las variables dependientes en función de los servicios públicos y las perturbaciones. Después de esto, el cuarto paso es limpiar nuestros datos simulados; definimos valores iniciales para optimy unir las variables dependientes e independientes en matrices. En quinto lugar, usamos optim para estimar realmente nuestro modelo, nombrando la estrategia de salida dentro de la iteración . Por último, los resultados del modelo se escriben en las matrices beta.qre y stder.qre .

Después de ejecutar este ciclo, podemos echar un vistazo rápido al valor promedio de nuestros coeficientes estimados ( β^ ) escribiendo:

aplicar (beta.qre, 2, media)

En esta convocatoria para postularse , estudiamos nuestra matriz de coeficientes de regresión en la que cada fila representa uno de los 100 experimentos de Monte Carlo, y cada columna representa uno de los cuatro coeficientes de regresión. Estamos interesados ​​en los coeficientes, por lo que escribimos 2 para estudiar las columnas y luego tomamos la media de las columnas. Si bien sus resultados pueden diferir un poco de lo que se imprime aquí, particularmente si no configuró la semilla para que sea la misma, la salida debería verse así:

[1] 1.0037491 1.0115165 1.0069188 0.9985754

En resumen, los cuatro coeficientes estimados están cerca de 1, lo cual es bueno porque 1 es el valor poblacional de cada uno. Si quisiéramos automatizar un poco más nuestros resultados para indicarnos el sesgo en los parámetros, podríamos escribir algo como esto:

desviarse <- barrido (beta.qre, 2, c (1,1,1,1))

colMeans (desviarse)

En la primera línea, usamos el comando de barrido , que barre (o resta) la estadística de resumen de nuestra elección de una matriz. En nuestro caso, beta.qre es nuestra matriz de estimaciones de coeficientes en 100 simulaciones. El argumento 2 que ingresamos por separado indica aplicar la estadística por columna (por ejemplo, por coeficiente) en lugar de por fila (lo que habría sido por experimento). Por último, en lugar de enumerar una estadística empírica que queremos restar, simplemente enumeramos los parámetros de población verdaderos para restar. En la segunda línea, calculamos el sesgo tomando la desviación media por column, o por parámetro. Nuevamente, la producción puede variar con diferentes semillas y generadores de números, pero en general debería indicar que los valores promedio no están lejos de los valores reales de la población. Todas las diferencias son pequeñas:

[1] 0,003749060 0,011516459 0,006918824 -0,001424579

Otra cantidad que vale la pena es el error absoluto medio, que nos dice cuánto difiere una estimación del valor de la población en promedio. Esto es un poco diferente en el sentido de que las sobreestimaciones y las subestimaciones no pueden desaparecer (como podría ocurrir con el cálculo del sesgo). Un estimador insesgado aún puede tener una gran varianza de error y un gran error medio absoluto. Como ya definimos desviarnos antes, ahora solo necesitamos escribir:

colMeans (abs (desviado))

Nuestra salida muestra pequeños errores absolutos promedio:

[1] 0.07875179 0.08059979 0.07169820 0.07127819

Para tener una idea de cuán bueno o malo es este rendimiento, deberíamos volver a ejecutar este experimento de Monte Carlo usando otro tratamiento para comparar. Para una comparación simple, podemos preguntar qué tan bien funciona este modelo si aumentamos el tamaño de la muestra. Presumiblemente, nuestro error absoluto medio disminuirá con una muestra más grande, por lo que en cada experimento podemos simular una muestra de 5000 en lugar de 1000. Para hacer esto, vuelva a ejecutar todo el código de esta sección del capítulo, pero reemplace una sola línea. Al definir el número de casos, la tercera línea después del inicio de la sección. 11.7.2 - en lugar de escribir n <-1000 , escriba en su lugar: n <-5000. Al final del programa, cuando se reportan los errores absolutos medios, verá que son más pequeños. Nuevamente, variarán de una sesión a otra, pero el resultado final debería verse así:

[1] 0.03306102 0.02934981 0.03535597 0.02974488

Como puede ver, los errores son más pequeños que con 1000 observaciones. Nuestro tamaño de muestra es considerablemente mayor, por lo que esperamos que este sea el caso. Esto nos da una buena comparación.

Con la finalización de este capítulo, ahora debería tener el kit de herramientas necesario para programar en R siempre que su investigación tenga necesidades únicas que no puedan ser abordadas por comandos estándar, o incluso por paquetes aportados por el usuario. Con la finalización de este libro, debería tener una idea del amplio alcance de lo que R puede hacer por los analistas políticos, desde la gestión de datos hasta los modelos básicos y la programación avanzada. Una verdadera fortaleza de R es la flexibilidad que ofrece a los usuarios para abordar las complejidades y los problemas originales que puedan enfrentar. A medida que proceda a utilizar R en su propia investigación, asegúrese de seguir consultando los recursos en línea para descubrir nuevas y prometedoras capacidades a medida que surjan.

11.8 Problemas de práctica 1. Distribuciones de probabilidad: Calcule la probabilidad de cada uno de los siguientes eventos: un. Una variable estándar distribuida normalmente es mayor que 3.

1. Una variable distribuida normalmente con media 35 y desviación estándar 6 es mayor que 42.
2. X  <0,9 cuando x tiene la distribución uniforme estándar.
3. Bucles: Let h ( x , n ) = 1 + x +X2+ ⋯ +Xnorte=∑nortei = 0XI . Escriba un programa en R para calcular h (0. 9, 25) usando un bucle for .
4. Funciones: En la pregunta anterior, escribió un programa para calcular h ( x , n ) =∑nortei = 0XI para x  = 0. 9 y n  = 25. Convierta este programa en una función más general que toma dos argumentos, x y n , y devuelve h ( x ,  n ). Usando la función, determine los valores de h (0. 8, 30), h (0. 7, 50) y h (0. 95, 20).
5. Estimación de máxima verosimilitud. Considere un ejemplo aplicado de Signorino (1999) método probit estratégico multinomial. Descargue un subconjunto de disputas interestatales militarizadas del siglo XIX, el archivo war1800.dta en formato Stata , del Dataverse (consulte la página vii) o el contenido en línea de este capítulo (consulte la página 205). Estos datos provienen de fuentes como EUGene (Bueno de Mesquita y Lalman 1992) y el Proyecto Correlatos de Guerra (Jones et al. 1996). Programe una función de verosimilitud para un modelo como el que se muestra en la figura  11.5 y estime el modelo para estos datos reales. Los tres resultados son: guerra (codificado 1 si los países entraron en guerra), sq (codificado 1 si se mantuvo el status quo) y capit (codificado 1 si el país objetivo capituló). Suponga que U 1 (SQ) es impulsado por pacificadores (número de años desde que la díada estuvo en conflicto por última vez) y s\_wt\_re1 (puntaje S para la similitud política de los estados, ponderado por la región del agresor). U 1 ( W ) es una función de balanc(la capacidad militar del agresor en relación con la capacidad combinada de la díada). U 2 ( W ) es una función de una constante y un equilibrio . U 1 ( C ) es una constante y U 2 ( C ) es cero. un. Informe sus estimaciones y los errores estándar correspondientes.
6. ¿Qué coeficientes se distinguen estadísticamente de cero?
7. Bonificación: Dibuje la probabilidad de guerra predicha si balanc se manipula desde su mínimo de 0 hasta su máximo de 1, mientras que pacifistas y s\_wt\_re1 se mantienen en sus mínimos teóricos de 0 y -1, respectivamente. Simplemente cómico: si dibuja el mismo gráfico de probabilidades predichas, pero permite que balanc llegue hasta 5, realmente puede ilustrar el tipo de relación no monótona que permite este modelo. Sin embargo, recuerde que no desea interpretar resultados fuera del rango de sus datos. Esta es solo una ilustración divertida para practicar más.
8. Análisis y optimización de Monte Carlo: Replica el experimento en Signorino’s (1999) probit multinomial estratégico de la Sect. 11,7 . ¿Hay mucha diferencia entre sus valores promedio estimados y los valores de la población? ¿Obtiene errores medios absolutos similares a los informados en esa sección? ¿Cómo se comparan sus resultados cuando prueba los siguientes tratamientos? un. Suponga que disminuye la desviación estándar de x1 , x2 , x3 y x4 a 0.5 (en contraposición al tratamiento original que suponía una desviación estándar de 1). ¿Mejoran o empeoran sus estimaciones? ¿Qué sucede si reduce aún más la desviación estándar de estos cuatro predictores, a 0,25? ¿Qué patrón general ve y por qué lo ve? ( Recuerde: todo lo demás sobre estos tratamientos, como el tamaño de la muestra y el número de experimentos, debe ser el mismo que el del experimento de control original. De lo contrario, todo lo demás no se mantiene igual).
9. Bonificación: ¿Qué sucede si tiene una variable omitida en el modelo? Cambie su función de probabilidad logarítmica para excluir x4 de su procedimiento de estimación, pero continúe incluyendo x4 cuando simule los datos en el ciclo for . Dado que solo estima los coeficientes para x1 , x2 y x3 , ¿cómo se comparan sus estimaciones en este tratamiento con el tratamiento original en términos de sesgo y error absoluto medio?

Notas al pie 1 . Véase también: Signorino (2002) y Signorino y Yilmaz (2003).

2 . Se pueden cargar otras distribuciones a través de varios paquetes. Por ejemplo, otra distribución útil es la normal multivariante. Al cargar la biblioteca MASS , un usuario puede tomar muestras de la distribución normal multivariante con el comando mvrnorm . Incluso de manera más general, el paquete mvtnorm permite al usuario calcular probabilidades t multivariadas normales y multivariadas , cuantiles, desviaciones aleatorias y densidades.

3 . Formalmente, la composición de esta lista se basa en una distribución uniforme estándar, que luego se convierte a la distribución que queramos utilizando una función de cuantiles.

4 . Recuerde del Cap. 1 que la función módulo nos da el resto de la división.

5 . Ver Nocedal y Wright (1999) para una revisión de cómo funcionan estas técnicas.

6 . Cuando se utiliza la estimación de máxima verosimilitud, como alternativa al uso de optim es utilizar el paquete maxLik . Esto puede ser útil específicamente para la máxima probabilidad, aunque optim tiene la ventaja de poder maximizar o minimizar también otros tipos de funciones, cuando sea apropiado.

7 . Para ver la versión original completa del programa, consulte <http://hdl.handle.net/1902.1/16781> . Tenga en cuenta que aquí usamos la familia de objetos S3 , pero el código original usa la familia S4 . El programa original también considera funciones alternativas de utilidad para votantes además de la función de proximidad cuadrática que se enumera aquí, como una función de proximidad absoluta y el modelo direccional de utilidad para votantes (Rabinowitz y Macdonald 1989).

8 . Si prefiere utilizar el sistema de objetos S4 para este ejercicio, lo siguiente que tendríamos que hacer es definir la clase de objeto antes de definir la función. Si quisiéramos llamar a nuestra clase de objeto simulación , escribiríamos: setClass (“simulación,” representación (resultadoA = “matriz,” resultadoD = “matriz,” bestResponseA = “numérico,” bestResponseD = “numérico,” equilibrioA = “carácter,” equilibriumD =" carácter ")) . El sistema de objetos S3 no requiere este paso, como veremos cuando creemos objetos de nuestra clase autodefinida game.simulation .

9 . Por ejemplo, en la frase resultA = resultA , el término a la izquierda del signo igual indica que este término en la lista se llamará resultA , y el término a la derecha llama al objeto de este nombre de la función para llenar este espacio .

10 . El código diferiría ligeramente para el sistema de objetos S4 . Si definimos nuestra clase de simulación como se describe en la nota al pie anterior, aquí reemplazaríamos la definición de resultado de la siguiente manera: resultado <-new (“simulación,” resultadoA = resultadoA, resultadoD = resultadoD, bestResponseA = bestResponseA, bestResponseD = bestResponseD, equilibriumA = equilibrio A, equilibrio D = equilibrio D) . Reemplazar el comando list con el nuevo comando asigna la salida a nuestra clase de simulación y llena las distintas ranuras en un solo paso. Esto significa que podemos omitir el paso adicional de usar el comando de clase , que necesitábamos cuando usamos el sistema S3 .

11 . Si, en cambio, hubiéramos guardado treatment.1 como un objeto de simulación S4 como se describe en las notas al pie anteriores, el comando para llamar a un atributo específico sería en su lugar: [treatment.1@equilibriumA](mailto:treatment.1@equilibriumA) .

12 . Los lectores interesados ​​en realizar una investigación como esta en su propio trabajo deben leer sobre el paquete de juegos , que fue desarrollado para estimar este tipo de modelo.

Material suplementario 318886\_1\_En\_11\_MOESM1\_ESM.zip (392 kb) Dataverse (2,154 KB) Referencias Alvarez RM, Levin I, Pomares J, Leiras M (2013) Votar de forma segura y sencilla: el impacto del voto electrónico en la percepción ciudadana. Polit Sci Res Methods 1 (1): 117-137 CrossRefGoogle Académico Bates D, Maechler M, Bolker B, Walker S (2014) lme4 : modelos lineales de efectos mixtos que utilizan Eigen y S4. Versión del paquete R 1.1-7. <http://www.CRAN.R-project.org/package=lme4> Becker RA, Cleveland WS, Shyu MJ (1996) El diseño visual y el control de la pantalla Trellis. J Comput Graph Stat 5 (2): 123-155 Google Académico Beniger JR, Robyn DL (1978) Gráficos cuantitativos en estadística: una breve historia. Am Stat 32 (1): 1–11 zbMATHGoogle Académico Berkman M, Plutzer E (2010) Evolución, creacionismo y la batalla por controlar las aulas de Estados Unidos. Cambridge University Press, Nueva York CrossRefGoogle Académico Black D (1948) Sobre el fundamento de la toma de decisiones en grupo. J Polit Econ 56 (1): 23–34 CrossRefGoogle Académico Black D (1958) La teoría de los comités y las elecciones. Cambridge University Press, Londres zbMATHGoogle Académico Cuadro GEP, Tiao GC (1975) Análisis de intervenciones con aplicaciones a problemas económicos y ambientales. Asociación J Am Stat 70: 70–79 CrossRefMathSciNetzbMATHGoogle Académico Box GEP, Jenkins GM, Reinsel GC (2008) Análisis de series de tiempo: previsión y control, 4ª ed. Wiley, Hoboken, Nueva Jersey CrossRefzbMATHGoogle Académico Box-Steffensmeier JM, Freeman JR, Hitt MP, Pevehouse JCW (2014) Análisis de series de tiempo para las ciencias sociales. Cambridge University Press, Nueva York CrossRefGoogle Académico Brambor T, Clark WR, Golder M (2006) Comprensión de los modelos de interacción: mejora de los análisis empíricos. Polit Anal 14 (1): 63–82 CrossRefGoogle Académico Brandt PT, Freeman JR (2006) Avances en el modelado de series de tiempo bayesiano y el estudio de la política: pruebas teóricas, pronósticos y análisis de políticas. Polit Anal 14 (1): 1–36 CrossRefGoogle Académico Brandt PT, Williams JT (2001) Un modelo lineal autorregresivo de Poisson: el modelo de Poisson AR (p). Polit Anal 9 (2): 164–184 CrossRefGoogle Académico Brandt PT, Williams JT (2007) Múltiples modelos de series temporales. Sage, Thousand Oaks, CA Google Académico Bueno de Mesquita B, Lalman D (1992) Guerra y razón. Prensa de la Universidad de Yale, New Haven Google Académico Carlin BP, Louis TA (2009) Métodos bayesianos para el análisis de datos. Chapman & Hall / CRC, Boca Ratón, FL zbMATHGoogle Académico Libro de cocina de gráficos Chang W (2013) R. O’Reilly, Sebastopol, CA Google Académico Cleveland WS (1993) Visualización de datos. Prensa Hobart, Sebastopol, CA Google Académico Cowpertwait PSP, Metcalfe AV (2009) Serie temporal introductoria con R. Springer, Nueva York zbMATHGoogle Académico Cryer JD, Chan KS (2008) Análisis de series de tiempo con aplicaciones en R, 2ª ed. Springer, Nueva York CrossRefzbMATHGoogle Académico Downs A (1957) Una teoría económica de la democracia. Harper and Row, Nueva York Google Académico Eliason SR (1993) Estimación de máxima verosimilitud: lógica y práctica. Sage, Thousand Oaks, CA Google Académico Enders W (2009) Series de tiempo econométricas aplicadas, 3ª ed. Wiley, Nueva York Google Académico Fitzmaurice GM, Laird NM, Ware JH (2004) Análisis longitudinal aplicado. Wiley-Interscience, Hoboken, Nueva Jersey zbMATHGoogle Académico Fogarty BJ, Monogan JE III (2014) Modelado de datos de recuento de series de tiempo: los desafíos únicos que enfrentan los estudios de comunicación política. Soc Sci Res 45: 73–88 CrossRefGoogle Académico Gelman A, Hill J (2007) Análisis de datos utilizando modelos de regresión y multinivel / jerárquicos. Cambridge University Press, Nueva York Google Académico Gelman A, Carlin JB, Stern HS, Rubin DB (2004) Análisis de datos bayesianos, 2ª ed. Chapman & Hall / CRC, Boca Ratón, FL zbMATHGoogle Académico Gibney M, Cornett L, Wood R, Haschke P (2013) Escala de terror político, 1976–2012. Obtenido el 27 de diciembre de 2013 del sitio web de la escala de terror político: <http://www.politicalterrorscale.org> Gill J (2001) Modelos lineales generalizados: un enfoque unificado. Sage, Thousand Oaks, CA Google Académico Gill J (2008) Métodos bayesianos: un enfoque de las ciencias sociales y del comportamiento, 2ª ed. Chapman & Hall / CRC, Boca Ratón, FL zbMATHGoogle Académico Granger CWJ (1969) Investigación de relaciones causales mediante modelos econométricos y métodos espectrales cruzados. Econometrica 37: 424–438 CrossRefGoogle Académico Granger CWJ, Newbold P (1974) Regresiones espurias en econometría. J Econ 26: 1045–1066 zbMATHGoogle Académico Gujarati DN, Porter DC (2009) Econometría básica, 5ª ed. McGraw-Hill / Irwin, Nueva York Google Académico Halley E (1686) Un relato histórico de los vientos alisios y monzones, observables en los mares entre y cerca de los trópicos, con un intento de asignar la causa física de dichos vientos. Philos Trans 16 (183): 153–168 CrossRefGoogle Académico Hamilton JD (1994) Análisis de series de tiempo. Prensa de la Universidad de Princeton, Princeton, Nueva Jersey zbMATHGoogle Académico Hanmer MJ, Kalkan KO (2013) Detrás de la curva: aclarando el mejor enfoque para calcular las probabilidades pronosticadas y los efectos marginales a partir de modelos de variables dependientes limitadas. Am J Polit Sci 57 (1): 263–277 CrossRefGoogle Académico Honaker J, King G, Blackwell M (2011) Amelia II: un programa para datos faltantes. J Stat Softw 45 (7): 1–47 CrossRefGoogle Académico Hotelling H (1929) Estabilidad en competencia. Econ J 39 (153): 41–57 CrossRefGoogle Académico Huber PJ (1967) El comportamiento de las estimaciones de máxima verosimilitud en condiciones no estándar. En: LeCam LM, Neyman J (eds) Actas del quinto simposio de Berkeley sobre estadística matemática y probabilidad, volumen 1: estadística University of California Press, Berkeley, CA Google Académico Iacus SM, King G, Porro G (2009) cem : software para emparejamiento exacto aproximado. J Stat Softw 30 (9): 1–27 CrossRefGoogle Académico Iacus SM, King G, Porro G (2011) Métodos de emparejamiento multivariante que limitan el desequilibrio monótono. J Am Stat Assoc 106 (493): 345–361 CrossRefMathSciNetzbMATHGoogle Académico Iacus SM, King G, Porro G (2012) Inferencia causal sin verificación de saldo: coincidencia exacta aproximada. Polit Anal 20 (1): 1–24 CrossRefGoogle Académico Imai K, van Dyk DA (2004) Inferencia causal con regímenes de tratamiento general: generalización de la puntuación de propensión. J Am Stat Assoc 99 (467): 854–866 CrossRefMathSciNetzbMATHGoogle Académico Jones DM, Bremer SA, Singer JD (1996) Disputas interestatales militarizadas, 1816-1992: fundamento, reglas de codificación y patrones empíricos. Confl Manag Peace Sci 15 (2): 163–213 CrossRefGoogle Académico Kastellec JP, Leoni EL (2007) Uso de gráficos en lugar de tablas en ciencias políticas. Perspect Polit 5 (4): 755–771 CrossRefGoogle Académico Keele L, Kelly NJ (2006) Modelos dinámicos para teorías dinámicas: los entresijos de las variables dependientes rezagadas. Polit Anal 14 (2): 186–205 CrossRefGoogle Académico King G (1989) Metodología política unificadora. Cambridge University Press, Nueva York Google Académico King G, Honaker J, Joseph A, Scheve K (2001) Analizando datos incompletos de ciencia política: un algoritmo alternativo para la imputación múltiple. Am Polit Sci Rev 95 (1): 49–69 Google Académico Koyck LM (1954) Rezagos distribuidos y análisis de inversiones. Holanda Septentrional, Amsterdam Google Académico Laird NM, Fitzmaurice GM (2013) Modelado de datos longitudinal. En: Scott MA, Simonoff JS, Marx BD (eds) El manual Sage de modelado multinivel. Sage, Thousand Oaks, CA Google Académico LaLonde RJ (1986) Evaluación de las evaluaciones econométricas de programas de formación con datos experimentales. Am Econ Rev 76 (4): 604–620 Google Académico Little RJA, Rubin DB (1987) Análisis estadístico con datos faltantes, 2ª ed. Wiley, Nueva York zbMATHGoogle Académico Long JS (1997) Modelos de regresión para variables dependientes categóricas y limitadas. Sage, Thousand Oaks, CA zbMATHGoogle Académico Lowery D, Gray V, Monogan JE III (2008) La construcción de comunidades de interés: distinguiendo modelos ascendentes y descendentes. J Polit 70 (4): 1160–1176 CrossRefGoogle Académico Lütkepohl H (2005) Nueva introducción al análisis de múltiples series de tiempo. Springer, Nueva York CrossRefzbMATHGoogle Académico Martin AD, Quinn KM, Park JH (2011) MCMCpack : Markov chain Monte Carlo en R. J Stat Softw 42 (9): 1–21 CrossRefGoogle Académico Mátyás L, Sevestre P (eds) (2008) La econometría de los datos de panel: fundamentos y desarrollos recientes en la teoría y la práctica, 3ª ed. Springer, Nueva York zbMATHGoogle Académico McCarty NM, Poole KT, Rosenthal H (1997) Redistribución de la renta y realineamiento de la política estadounidense. Estudios del instituto empresarial estadounidense sobre la comprensión de la desigualdad económica. AEI Press, Washington, DC Google Académico Monogan JE III (2011) Análisis de datos de panel. En: Badie B, Berg-Schlosser D, Morlino L (eds) Enciclopedia internacional de ciencia política. Sage, Thousand Oaks, CA Google Académico Monogan JE III (2013a) Un caso para registrar estudios de resultados políticos: una aplicación en las elecciones a la Cámara de 2010. Polit Anal 21 (1): 21–37 CrossRefGoogle Académico Monogan JE III (2013b) Colocación estratégica del partido con un electorado dinámico. J Theor Polit 25 (2): 284–298 CrossRefGoogle Académico Nocedal J, Wright SJ (1999) Optimización numérica. Springer, Nueva York CrossRefzbMATHGoogle Académico Owsiak AP (2013) Democratización y acuerdos fronterizos internacionales. J Polit 75 (3): 717–729 CrossRefGoogle Académico Peake JS, Eshbaugh-Soha M (2008) El impacto en el establecimiento de la agenda de los principales discursos televisivos presidenciales. Polit Commun 25: 113-137 CrossRefGoogle Académico Petris G, Petrone S, Campagnoli P (2009) Modelos lineales dinámicos con R. Springer, Nueva York CrossRefzbMATHGoogle Académico Pfaff B (2008) Análisis de series temporales integradas y cointegradas con R, 2ª ed. Springer, Nueva York CrossRefzbMATHGoogle Académico Playfair W (1786/2005) En: Wainer H, Spence I (eds) Atlas comercial y político y breviario estadístico. Cambridge University Press, Nueva York Google Académico Poe SC, Tate CN (1994) Represión de los derechos humanos a la integridad personal en la década de 1980: un análisis global. Am Polit Sci Rev 88 (4): 853–872 CrossRefGoogle Académico Poe SC, Tate CN, Keith LC (1999) Revisión de la represión del derecho humano a la integridad personal: un estudio internacional transnacional que abarca los años 1976-1993. Int Stud Q 43 (2): 291–313 CrossRefGoogle Académico Poole KT, Rosenthal H (1997) Congreso: una historia político-económica de la votación nominal. Oxford University Press, Nueva York Google Académico Poole KT, Lewis J, Lo J, Carroll R (2011) Escala de votos nominales con wnominate en R. J Stat Softw 42 (14): 1–21 CrossRefGoogle Académico Rabinowitz G, Macdonald SE (1989) Una teoría direccional de la votación por cuestiones. Am Polit Sci Rev 83: 93-121 CrossRefGoogle Académico Robert CP (2001) La elección bayesiana: de los fundamentos de la teoría de la decisión a la implementación computacional, 2ª ed. Springer, Nueva York zbMATHGoogle Académico Rubin DB (1987) Imputación múltiple por falta de respuesta en encuestas. Wiley, Nueva York CrossRefzbMATHGoogle Académico Rubin DB (2006) Muestreo emparejado para efectos causales. Cambridge University Press, Nueva York CrossRefzbMATHGoogle Académico Scott MA, Simonoff JS, Marx BD (eds) (2013) El manual Sage de modelado multinivel. Sage, Thousand Oaks, CA Google Académico Sekhon JS, Grieve RD (2012) Un método de emparejamiento para mejorar el equilibrio de covariables en los análisis de rentabilidad. Health Econ 21 (6): 695–714 CrossRefGoogle Académico Shumway RH, Stoffer DS (2006) Análisis de series de tiempo y sus aplicaciones con ejemplos de R, 2ª ed. Springer, Nueva York zbMATHGoogle Académico Signorino CS (1999) Interacción estratégica y análisis estadístico del conflicto internacional. Am Polit Sci Rev 93: 279–297 CrossRefGoogle Académico Signorino CS (2002) Estrategia y selección en relaciones internacionales. Int Interact 28: 93-115 CrossRefGoogle Académico Signorino CS, Yilmaz K (2003) Especificación errónea estratégica en modelos de regresión. Am J Polit Sci 47: 551–566 CrossRefGoogle Académico Sims CA, Zha T (1999) Bandas de error para respuestas de impulso. Econometrica 67 (5): 1113–1155 CrossRefMathSciNetzbMATHGoogle Académico Singh SP (2014a) Funciones de pérdida de utilidad lineales y cuadráticas en la investigación del comportamiento electoral. J Theor Polit 26 (1): 35–58 CrossRefGoogle Académico Singh SP (2014b) No todos los ganadores de las elecciones son iguales: satisfacción con la democracia y la naturaleza del voto. Eur J Polit Res 53 (2): 308–327 CrossRefGoogle Académico Singh SP (2015) Voto obligatorio y cálculo de decisiones de participación. Polit Stud 63 (3): 548–568 CrossRefGoogle Académico Tufte ER (2001) La presentación visual de información cuantitativa, 2ª ed. Prensa gráfica, Cheshire, CT Google Académico Tukey JW (1977) Análisis de datos exploratorios. Addison-Wesley, Reading, PA zbMATHGoogle Académico Wakiyama T, Zusman E, Monogan JE III (2014) ¿Puede sostenerse una transición energética baja en carbono en el Japón posterior a Fukushima? Evaluación de los diferentes impactos de los choques exógenos. Política energética 73: 654–666 CrossRefGoogle Académico Wei WWS (2006) Análisis de series de tiempo: métodos univariados y multivariados, 2ª ed. Pearson, Nueva York zbMATHGoogle Académico White H (1980) Un estimador de matriz de covarianza consistente con heterocedasticidad y una prueba directa de heterocedasticidad. Econometrica 48 (4): 817–838 CrossRefMathSciNetzbMATHGoogle Académico Yau N (2011) Visualice esto: la guía FlowingData para diseño, visualización y estadísticas. Wiley, Indianápolis Google Académico

* ADORNO, Theodor W. “LA PERSONALIDAD AUTORITARIA,” Paidós, Bs.As., 1969.
* ALIGHIERI, Dante “DE LA MONARQUIA,” Losada, Bs.As.
* ALBERIONI et al. “L’ATTIVISTA DI PARTITO,” Bolonia, 1967.
* ALMOND, G. y COLEMAN “THE POLITICS OF THE DEVELOPING AREAS,” Princeton University Press, 1960.
* ALMOND, G.A. y VERBA, Sidney “THE CIVIC CULTURE,” Princeton University Press, 1963.
* ALMOND, G.A. y POWELL, G.B. “POLITICA COMPARADA,” Paidós, Bs.As., 1972.
* ALTHUSSER, Louis “PARA LEER EL CAPITAL,” Siglo XXI, México, 1969.
* AMIN, Samir “SOBRE EL DESARROLLO DESIGUAL DE LAS FORMACIONES SOCIALES,” Anagrama, Barcelona, 1976.
* AMIN, Samir “L’ACCUMULATION A L’ECHELLE MONDIALE,” Editions Anthropos, París,
* ANDERSON, Perry “CONSIDERACIONES SOBRE EL MARXISMO OCCIDENTAL,” Siglo XXI, México, 1990.
* APONTE, Antonio “LA ECONOMIA DE LOS PAISES SOCIALISTAS,” Salvat ed., Barcelona,
* ARENDT, Hannah “LOS ORIGENES DEL TOTALITARISMO,” Alianza ed., Madrid, 1981-1982, 2 vol.
* APTER, David E. “POLITICA DE LA MODERNIZACION,” Paidós, Bs.As., 1972.
* ARISTOTELES “LA POLITICA,” Editora Nacional, Madrid, 1977.
* ARNOLETTO, Eduardo J. “APROXIMACION A LA CIENCIA POLITICA,” Artesol ed., Córdoba,
* ARON, Raymond “EL OPIO DE LOS INTELECTUALES,” Siglo XX, Bs.As., 1968.
* ARON, Raymond “DEMOCRACIA Y TOTALITARISMO,” Seix, Barcelona, 1971.
* ARON, Raymond “PAZ Y GUERRA ENTRE LAS NACIONES,” Alianza ed., Madrid, 1984.
* ARON, Raymond “REPUBLIQUE IMPERIALE: LES ETATS-UNIS DANS LE MONDE (1945- 1972),” Calmann-Lévy, París, 1973.
* “BREVE DICCIONARIO POLITICO,” Ed. Progreso, Moscú, 1983.
* BALANDIER, George “ANTROPOLOGIA POLITICA,” Península, Barcelona, 1969.- BARAN, Paul y SWEEZY, Paul “EL CAPITALISMO MONOPOLISTA. UN ENSAYO SOBRE LA SOCIEDAD INDUSTRIAL AMERICANA,” Siglo XXI, México, 1968.
* BARRACLOUGH, Geofrey “UNE INTRODUCTION A L’HISTOIRE CONTEMPORAINE,” Ed. Stock, París, 1964.
* BELL, Daniel “FIN DE LAS IDEOLOGIAS,” Tecnos, Madrid, 1964.
* BELL, Daniel “THE END OF IDEOLOGY: ON THE EXHAUSTION OF POLITICAL IDEAS IN THE FIFTIES,” New York, 1960.
* BENDIX, R. “NATION-BUILDING AND CITIZENSHIP,” John Wiley, New York, 1964.
* BERTALANFFY, Ludwig von “TEORIA GENERAL DE LOS SISTEMAS,” FCE, México, 1981.
* BENOIST, Alain de “DEMOCRATIE: LE PROBLEME,” Le Labyrinthe, París, 1985.
* BEYME, Klaus von “TEORIAS POLITICAS CONTEMPORANEAS - UNA INTRODUCCION,” Instituto de Estudios Políticos, Madrid, 1977.
* BOBBIO, N. et al. “DICCIONARIO DE POLITICA,” Siglo XXI, México, 1986.
* BOBBIO, Norberto “SAGGI SULLA SCIENZA POLITICA IN ITALIA,” Bari, 1969.
* BOBBIO, Norberto “IL FUTURO DELLA DEMOCRAZIA. UNA DIFESA DELLE REGOLE DEL GIOCO,” Einaudi ed., Torino, 1984.
* LE BON, Gustave “PSICOLOGIA DE LAS MULTITUDES,” Ed. Albatros, Bs.As., 1978.
* BRZEZINSKI, Zbigniew “IDEOLOGIA Y PODER EN LA POLITICA SOVIETICA,” Paidós, Bs.As., 1968.
* BRAILLARD, Philippe “THEORIE DES SYSTEMES ET RELATIONS INTERNA-TIONALES,” Ed. Bruylant, Bruselas, 1977.
* BRAILLARD, P. y DE SENARCLENS, P. “EL IMPERIALISMO,” FCE, México, 1982.
* BRUNSCHWIG, H. “LE PARTAGE DE L’AFRIQUE NOIRE,” Flammarion, París, 1971.
* CARDOZO, F.H. y FALETTO, E. "DEPENDENCIA Y DESARROLLO EN AMERICA LATINA, Siglo XXI, México, 1969.
* CARTWRIGHT, Dorwin y ZANDER, Alvin “GROUP DYNAMICS: RESEARCH AND THEORY,” Ed. Harper and Row, 1962.
* CASSIRER, Ernest “EL MITO DEL ESTADO,” FCE, México, 1968. CESAREO, Vincenzo et al. “LA CULTURA DELL’ITALIA CONTEMPORANEA. TRASFORMAZIONE DEI MODELLI DI COMPORTAMENTO E IDENTITA SOCIALE,” Ed. Fondazione Giovanni Agnelli, Torino, 1990.
* CHATELET, F., DUHAMEL, O. y PISIER, E. “DICTIONNAIRE DES OEUVRES POLITIQUES,” P.U.F., París, 1989.- CHEVALIER, J.J. “LOS GRANDES TEXTOS POLITICOS - DESDE MAQUIAVELO A NUESTROS DIAS,” Aguilar, Madrid, 1979.
* COPLIN, W.D. "INTRODUCTION TO INTERNATIONAL POLITICS. A THEORETICAL OVERVIEW, Chicago ,1971.
* CROZIER, Michel “LE PHENOMENE BUREAUCRATIQUE,” Seuil, París, 1964.
* DENQUIN, Jean-Marie “SCIENCE POLITIQUE,” P.U.F., París, 1991.
* DEUTSCH, K.W. “NATIONALISM AND SOCIAL COMMUNICATION. AN INQUIRY INTO THE FOUNDATIONS OF NATIONALITY,” M.I.T. Press, Mass., 1953.
* DEUTSCH, K. et al. “POLITICAL COMMUNITY AND THE NORTH ATLANTIC AREA. INTERNATIONAL ORGANIZATION IN THE LIGHT OF HISTORICAL EXPERIENCE,” Princeton, 1957.
* DEUTSCH, Karl “POLITICA Y GOBIERNO,” FCE, México, 1976.
* DEUTSCH, Karl “LOS NERVIOS DEL GOBIERNO,” FCE, México, 1985.
* DEUTSCH, Morton y KRAUSS, Robert “THEORIES IN SOCIAL PSYCHOLOGY,” Basic Books, Inc., 1965.
* DIAMANT, A. “THE NATURE OF POLITICAL DEVELOPMENT” en Finkle y Gable “POLITICAL DEVELOPMENT AND SOCIAL CHANGE,” John Wiley, New York, 1966.
* DJILAS, Milovan “LA NUEVA CLASE,” Sudamericana, Bs.As., 1965.
* DRAPER, T. “ABUSE OF POWER,” Secker and Warburg, London, 1966.
* DURKHEIM, E. "DE LA DIVISION DEL TRABAJO SOCIAL, Schapire, Bs.As, 1967.
* ———— “EL SUICIDIO,” Schapire, Bs.As., 1965.
* ———— “LAS FORMAS ELEMENTALES DE LA VIDA RELIGIOSA,” Schapire, Bs.As., 1968.
* EASTON ,D. y DENNIS, J. “CHILDREN IN THE POLITICAL SYSTEM,” New York, 1969.
* EASTON, David "ESQUEMA PARA EL ANALISIS POLITICO, Amorrortu, Bs.As., 1969.
* ECKSTEIN, H. y APTER, D. “COMPARATIVE POLITICS. A READER,” New York, 1963.
* ECKSTEIN, H. “DIVISION AND COHESION IN DEMOCRACY - A STUDY OF NORWAY,” Princeton University Press, 1966.
* EISENSTADT, S.N. “MODERNIZACION, MOVIMIENTOS DE PROTESTA Y CAMBIO SOCIAL,” Amorrortu, Bs.As., 1969.
* EMMANUEL, Arghiri "L’ECHANGE INEGAL. ESSAI SUR LES ANTAGONISMES DANS LES RAPPORTS ECONOMIQUES INTERNATIONAUX, Maspero, París, 1969.
* FEJTÖ, François “LA SOCIAL-DEMOCRATIE QUAND MEME,” Ed. Robert Laffont, París, 1980. - FESTINGER, Leo “A THEORY OF COGNITIVE DISSONANCE,” Row, Peterson and Co., 1957.
* FRANK, André Gunder “CAPITALISMO Y SUBDESARROLLO EN AMERICA LATINA,” Siglo XXI, México, 1970.
* FRENCH, John R.P. Jr. “A FORMAL THEORY OF SOCIAL POWER” en Cartwright y Zander: “GROUPS DINAMICS: RESEARCH AND THEORY,” Ed. Harper and Row, 1962.
* FREUD, Sigmund “OBRAS COMPLETAS” , Tomos II y III, Ed. Biblioteca Nueva, Madrid, 1973.
* FRIEDRICH, Carl “EL HOMBRE Y EL GOBIERNO: UNA TEORIA EMPIRICA DE LA POLITICA,” Tecnos, Madrid, 1968.
* FRIEDRICH, C. y BRZEZINSKI, Z. “DICTADURA TOTALITARIA Y AUTOCRACIA,” Libera, Bs.As., 1975.
* FULBRIGHT, J.W. “THE ARROGANCE OF POWER,” Vintage Books, New York, 1966.
* FURTADO, Celso “DESARROLLO Y SUBDESARROLLO,” Eudeba, Bs.As., 1965.
* GALLAGER, John y ROBINSON, Ronald “AFRICA AND THE VICTORIANS. THE OFFICIAL MIND OF IMPERIALISM,” Ed. Macmillan, London, 1961.
* GARCIA PELAYO, Manuel “MITOS Y SIMBOLOS POLITICOS,” Taurus, Madrid, 1964.
* GENIAGE, Jean “L’EXPANSION COLONIALE DE LA FRANCE SOUS LA IIIe REPUBLIQUE (1871-1914),” Payot, París, 1968.
* GERMANI, Gino “POLITICA Y SOCIEDAD EN UNA EPOCA DE TRANSICION,” Paidós, Bs.As., 1965.
* GOLEMBIEWSKI, Robert “BEHAVIOR AND ORGANIZATION: ORGANIZATION AND METHODS AND THE SMALL GROUP,” Rand McNally and Co., 1962.
* GORI, U., BRUSCHI, A., ATTINA, F. “RELAZIONI INTERNAZIONALI. METODI E TECNICHE DI ANALISI,” Milán, 1974.
* GRAMSCI, Antonio “NOTAS SOBRE MAQUIAVELO, LA POLITICA Y EL ESTADO,” Juan Pablos, México, 1975.
* GURVITCH, Georges “TRATADO DE SOCIOLOGIA,” Kapeluz, BS.As., 1963.
* —————– “LES CADRES SOCIAUX DE LA CONNAISSANCE,” PUF, París, 1966.
* HABERMAS, Jürgen “TEORIA Y PRAXIS,” Sur, Bs.As., 1967.
* —————- "TEORIA E PRASSI NELLA SOCIETA TECNOLOGICA, Bari, 1969.
* HARGROVE, E.C. “PRESIDENTIAL LEADERSHIP - PERSONALITY AND POLITICAL STYLE,” New York/London, 1966. - HAURIOU, A. “DERECHO CONSTITUCIONAL E INSTITUCIONES POLITICAS,” Ed. Ariel, Barcelona, 1971.
* HEARNSHAW, F.J.C. “HISTORIA DE LAS IDEAS POLITICAS,” Empresa Letras, Santiago de Chile, .
* HILFERDING, Rudolf “LE CAPITAL FINANCIER. ETUDE SUR LE DEVELOPMENT RECENT DU CAPITALISME,” Ed. du Minuit, París, 1970.
* HOBSON, J.A. “ESTUDIOS DEL IMPERIALISMO,” Alianza, Madrid, 1981.
* HOFFMAN, Stanley “GULLIVER EMPETRÉ. ESSAI SUR LA POLITIQUE ETRANGERE DES ETATS-UNIS,” Seuil, París, 1971.
* HOLT, R.T. y TURNER E. “THE METHODOLOGY OF COMPARATIVE RESEARCH,” New York, 1970.
* HOMANS, George C. “SOCIAL BEHAVIOR: ITS ELEMENTARY FORMS,” Harcourt, Brace and Word Inc., 1961.
* HORKHEIMER, Max y ADORNO, Theodor “DIALECTICA DEL ILUMINISMO,” Sur, Bs.As.,
* HOVLAND, Car I. et al. “COMMUNICATION AND PERSUATION: PSYCOLOGICAL STUDIES OF OPINION CHANGE,” Yale University Press, 1953.
* HULL, Clark L. “A BEHAVIOR SYSTEM,” Yale University Press, 1952.
* HUNTINGTON, S.P. y MOORE, C.H. “AUTHORITARIAN POLITICS IN MODERN SOCIETY,” New York, 1970.
* HUNTINGTON, S.P. “EL ORDEN POLITICO EN LAS SOCIEDADES EN CAMBIO,” Paidós, Bs.As., 1972.
* JAGUARIBE H. et al. “LA DEPENDENCIA POLITICO-ECONOMICA DE AMERICA LATINA,” Siglo XXI, México, 1971.
* JAGUARIBE-FURTADO-FALETTO-DITELLA-ESPARTACO-SUNKEL- CARDOSO “LA DOMINACION DE AMERICA LATINA,” Amorrortu, Bs.As., 1972.
* JAGUARIBE, Helio “SOCIEDAD, CAMBIO Y SISTEMA POLITICO,” Paidós, Bs.As., 1972.
* —————- “DESARROLLO POLITICO - SENTIDO Y CONDICIONES,” Paidós, Bs.As., 1972.
* —————- “AMERICA LATINA - REFORMA O REVOLUCION,” Paidós, Bs.As., 1972.
* —————- “O NOVO CENARIO INTERNACIONAL,” Ed. Guanabara, Río de Janeiro, 1986.
* JALEÉ, Pierre “L’IMPERIALISME EN 1970,” Maspero, París, 1973.
* JAMES. Emile “HISTORIA DEL PENSAMIENTO ECONOMICO,” Aguilar, Madrid, 1974. - JOUVENEL, Bertrand de “EL PODER,” Ed. Nacional, Madrid, 1974.
* 2a ed. KAPLAN, M.A. “SYSTEM AND PROCES IN INTERNATIONAL POLITICS,” New York,
* KEOHANE, R.O. y NYE, J.S. “TRANSNATIONAL RELATIONS IN WORLD POLITICS,” Harvard University Press, 1972.
* KNOLL, E. y McFADEN, J. “AMERICAN MILITARISM - 1970,” The Viking Press, New York,
* KORNHAUSER, William “ASPECTOS POLITICOS DE LA SOCIEDAD DE MASAS,” Amorrortu, Bs.As., 1969.
* LAGROYE, Jacques “SOCIOLOGIE POLITIQUE,” Presses de la F.N. des Sc. Po. & Dalloz, París,
* LANE, R. “POLITICAL IDEOLOGY,” New York, 1962.
* LAPLANCHE, J. y PONTALIS, J.B. “DICCIONARIO DE PSICOANALISIS,” Ed. Labor, Barcelona, 1974.
* LASSWELL, Harold D. “PSYCHOPATHOLOGY AND POLITICS,” Viking Press Inc., 1962.
* LERNER, D. "THE PASSING OF TRADITIONAL SOCIETY. MODERNIZING THE MIDDLE EAST, New York, 1958.
* LEWIN, Kurt “FIELD THEORY IN SOCIAL SCIENCE,” Dorwin Cartwright (Harper and Bros.),
* LIJPHART, A. “THE POLITICS OF ACCOMODATION. PLURALISM AND DEMOCRACY IN THE NETHERLANDS,” Berkeley, Los Angeles, 1968.
* LIPSET, Seymour Martin “EL HOMBRE POLITICO. LAS BASES SOCIALES DE LA POLITICA,” Eudeba ed., Bs.As., 1977.
* LISKA, George “IMPERIAL AMERICA. THE INTERNATIONAL POLITICS OF PRIMACY,” John Hopkins Press, Baltimore, 1967.
* LOCKE, John “ENSAYO SOBRE EL GOBIERNO CIVIL,” Aguilar, Madrid, 1981.
* LOPEZ, Mario Justo “INTRODUCCION A LOS ESTUDIOS POLITICOS,” Tomos I y II, Kapeluz, Bs.As., 1975.
* LUKACS, György “EL ASALTO A LA RAZON,” Grijalbo, México, 1976.
* LUXEMBURG, Rosa “LA ACUMULACION DEL CAPITAL,” Grijalbo, México, 1967.
* MAIMONIDES, M. “THE GUIDE OF THE PERPLEXED,” University of Chica-go Press, Chicago,
  + MANNHEIM, Karl “IDEOLOGIA Y UTOPIA,” Aguilar, Madrid, 1973.
* MAQUIAVELO, Nicolás “EL PRINCIPE,” Alianza ed., Madrid, 1981.
* ——————- “DISCURSOS SOBRE LA PRIMERA DECADA DE TITO LIVIO,” en “Obras,” Vergara, Barcelona, 1965.
* MARCH, James G. y SIMON, Herbert A. “ORGANIZATIONS,” John Wiley and Sons, 1962.
* MARX, Karl “LA IDEOLOGIA ALEMANA,” Grijalbo, México, 1969.
* MARX, Karl “ELEMENTOS FUNDAMENTALES PARA LA CRITICA DE LA ECONO- MIA POLITICA,” Siglo XXI, Madrid, 1972.
* ———- “TEORIAS SOBRE LA PLUSVALIA,” FCE, México, 1982.
* MARCUSE, Herbert “EL FIN DE LA UTOPIA,” Siglo XXI, México, 1968.
* —————- “EL HOMBRE UNIDIMENSIONAL,” Mortiz, México, 1970.
* MEEHAN, E.J. “PENSAMIENTO POLITICO CONTEMPORANEO,” Rev. de Occidente, Madrid,
* MERTON, Robert K. “TEORIA Y ESTRUCTURAS SOCIALES,” FCE, México, 1964.
* MICHELS, Robert “LOS PARTIDOS POLITICOS,” Amorrortu, Bs.As., 1969
* MILBRATH, Lester W. “POLITICAL PARTICIPATION,” Rand Mc Nally and Co., 1965.
* MOONEY, Alfredo y ARNOLETTO, Eduardo “CUESTIONES FUNDAMENTALES DE CIENCIA POLITICA,” Alveroni ed., Córdoba, 1993.
* MORGENTHAU, Hans “POLITICS AMONG NATIONS. THE STRUGGLE FOR POWER AND PEACE,” Knopf, New York, 1955.
* MORLINO, L. Comp. “GUIDE AGLI STUDI DI SCIENZE SOCIALI IN ITALIA - SCIENZA POLITICA,” Ed. Fond. G. Agnelli, Torino, 1989.
* MORO, Tomás “UTOPIA,” Bruguera, Barcelona, 1973.
* ORGANSKI, A.F.K. “THE STAGES OF POLITICAL DEVELOPMENT,” A. Knopf, New York,
* OSSOWSKI, Stanislaw “ESTRUCTURA DE CLASE Y CONCIENCIA SOCIAL,” Península, Barcelona, 1971.
* PARETO, Vilfredo “FORMA Y EQUILIBRIO SOCIALES,” Rev. de Occidente, Madrid, 1966.
* PARETI, Luigi et al. “HISTORIA DE LA HUMANIDAD - DESARROLLO CULTURAL Y CIENTIFICO,” Tomo II (UNESCO), Ed. Sudamericana, Bs.As., 1969.
* PARTRIDGE, P.H. “CONSENT AND CONSENSUS,” Londres, 1971.
* PARSONS, Talcott “EL SISTEMA SOCIAL,” Rev. de Occidente, Madrid, 1976. - —————- “ENSAYOS DE TEORIA SOCIOLOGICA,” Paidós, Bs.As., 1970.
* —————- “EL SISTEMA DE LAS SOCIEDADES MODERNAS,” Trillas México, 1974.
* PITKIN, H. “THE CONCEPT OF REPRESENTATION,” Berkeley, 1967.
* PINTO, A. “POLITICA Y DESARROLLO,” Ed. Universitaria, Santiago de Chile, 1972.
* PLATON “LA REPUBLICA,” UNAM, México, 1971.
* —— “LAS LEYES,” Inst. de Est. Políticos, Madrid, 1960.
* —— “EL POLITICO,” Inst. de Est. Políticos, Madrid, 1955.
* PUTNAM, R.D. "THE BELIEF OF POLITICIANS: IDEOLOGY, CONFLICT AND DEMOCRACY IN BRITAIN AND ITALY, London, 1973.
* PYE, L.W. “COMMUNICATIONS AND POLITICAL DEVELOPMENT,” Princeton, 1963.
* ——— “POLITICS, PERSONALITY AND NATION-BUIDING. BURMS’S SEARCH FOR IDENTITY,” Yale University Press, New Haven, 1966.
* ——— “ASPECTS OF POLITICAL DEVELOPMENT,” Little Brown, Boston, 1966.
* PYE, L.W. y VERBA, S. “POLITICAL CULTURE AND POLITICAL DEVELOP-MENT,” Princeton University Press, 1969.
* RIBEIRO, Darsy “EL DILEMA DE AMERICA LATINA,” Siglo XXI, México, 1971.
* RICHARDSON, Lewis “ARMS AND INSECURITY,” Quadrangle Press, Chicago, 1960.
* ROBINSON, R. “THE NON-EUROPEAN FOUNDATIONS OF EUROPEAN IMPERIALISM: SKETCH FOR A THEORY OF COLLABORATION,” Longman, London, 1972.
* ROUQUIÉ, Alain “EXTREMO OCCIDENTE. INTRODUCCION A AMERICA LATINA,” Emecé, Bs.As., 1991.
* ROSTOW, Walt “LAS ETAPAS DEL CRECIMIENTO ECONOMICO,” FCE, México, 1961.
* ROSTOW, W.W. “POLITICS AND THE STAGES OF GROWTH,” Cambridge, 1971.
* ROSENAU, J.N. “THE SCIENTIFIC STUDY OF FOREIGN POLICY,” New York, 1971.
* RUNCIMAN, W.G. “SOCIOLOGY IN ITS PLACE,” Cambridge, 1970.
* RUYER, Raymond “L’UTOPIE ET LES UTOPIES,” PUF, París, 1950.
* SABINE, G.H. “HISTORIA DE LA TEORIA POLITICA,” FCE, México, 1984.
* SANTOS, Theodoro dos “LA NUOVA DIPENDENZA,” Milán, 1971.
* SARTORI, G. “SISTEMI RAPPRESENTATIVI” en “DEMOCRAZIA E DEFINIZIO- NI,” Bolonia,
* ———– “LA POLITICA - LOGICA Y METODO EN LAS CIENCIAS SOCIA- LES,” FCE, México, 1984. - SCHLESINGER, A.M. “THE IMPERIAL PRESIDENCY,” Popular Library, New York, 1974.
* SCHUMPETER, Joseph “CAPITALISMO, SOCIALISMO Y DEMOCRACIA,” Agui- lar, México,
* —————— “IMPERIALISMO Y CLASES SOCIALES,” Tecnos, Ma- drid, 1965.
* SCHMITT, Carl “LEGALIDAD Y LEGITIMIDAD,” Aguilar, Madrid, . SAINT-SIMON “OEUVRES,” Anthropos, París, 1966.
* SIMON, Herbert A. “MODELS OF MAN: SOCIAL AND RATIONAL,” Wiley, New York, 1957.
* SKINNER, B.F. “SCIENCE AND HUMAN BEHAVIOR,” Free Press, New York, 1953.
* SOREL, Jean “REFLEXIONES SOBRE LA VIOLENCIA,” Alianza, Madrid, 1976.
* SUNKEL, O. y PAZ, P. “EL DESARROLLO LATINOAMERICANO Y LA TEORIA DEL DESARROLLO,” Siglo XXI, México, 1970.
* SWEEZY, Paul “ÉLITE DE PODER O CLASE DIRIGENTE?” Jorge Alvarez, Bs.As., . TUCKER, Robert “NATION OR EMPIRE? DEBATE OVER AMERICAN FOREIGN POLICY,” J. Hopkins Press, Baltimore, 1968.
* TZU, Sun “L’ART DE LA GUERRE,” Flammarion, París, 1972.
* URBANI, G. “LA POLITICA COMPARATA,” Bolonia, 1972.
* VERBA, Sidney “SMALL GROUPS AND POLITICAL BEHAVIOR,” Princeton University Press,
* VOEGELIN, Eric “NUEVA CIENCIA DE LA POLITICA,” Rialp, Madrid, 1968.
* VRANICKI, P., SUPEK, R. et al. “EL SOCIALISMO YUGOESLAVO ACTUAL,” Grijalbo ed., México, 1975.
* WEBER, Max “EL POLITICO Y EL CIENTIFICO,” Alianza, Madrid, 1967.
* ———- “ECONOMIA Y SOCIEDAD,” FCE, México, 1964.
* YARMOLINSKY, Adam “THE MILITARY ESTABLISHMENT. ITS IMPACT ON AMERICAN SOCIETY,” Harper & Row, New York, 1971.
* ZEITLIN, Irving “IDEOLOGIA Y TEORIA SOCIOLOGICA,” Amorrortu ed., Bs.As., 1973

Stock, J. H., and M. W. Watson. 2015. *Introduction to Econometrics, Third Update, Global Edition*. Pearson Education Limited.

1. Los nombres de estos archivos cambian a medida que se lanzan nuevas versiones de R. En el momento de esta impresión, los archivos respectivos son R-3.1.2-snowleopard.pkg o R-3.1.2-mavericks.pkg para varias versiones de Mac OS X y R-3.1.2-win.exe para Windows. A los usuarios de Linux les resultará más fácil de instalar desde una terminal. El código de terminal está disponible siguiendo el enlace \* Descargar \* R \* para Linux \* en la página de CRAN, luego eligiendo una distribución de Linux en la página siguiente y usando el código de terminal que aparece en la página resultante. En el momento de la impresión, se admiten Debian, varias formas de Red Hat, OpenSUSE y Ubuntu. [↑](#footnote-ref-27)
2. RStudio está disponible en [http://www.rstudio.com.](Http://www.rstudio.com) [↑](#footnote-ref-29)
3. The arrow (<-) is the traditional assignment operator, though a single equals sign (=) also can serve for assignments. [↑](#footnote-ref-32)
4. Una mirada más cercana al archivo en sí mostrará que nuestra línea de encabezado de nombres de variables en realidad tiene un elemento menos que cada línea de datos. Cuando este es el caso,**R** supone que el primer elemento de cada línea es un índice de observación. Dado que eso es cierto en este caso, nuestros datos se leen correctamente. [↑](#footnote-ref-41)
5. Otras figuras históricas particularmente clave en el desarrollo de medidas gráficas incluyen Halley (1686), Juega limpio (1786/2005) y Tukey (1977). Beniger y Robyn presentan una historia más completa (1978). [↑](#footnote-ref-44)
6. Nebraska y Carolina del Norte son observaciones faltantes del índice de Ranney. [↑](#footnote-ref-45)
7. El valor predeterminado de **las** es 0, que imprime etiquetas paralelas al eje. 1, nuestra elección aquí, los imprime horizontalmente. 2 imprime perpendicularmente al eje y 3 imprime verticalmente. [↑](#footnote-ref-46)
8. Sin embargo, como alternativa, si un usuario tuviera algún índice de tiempo en el marco de datos, se podría producir un gráfico similar escribiendo algo como: pres.energyEnergy,x=pres.energy$Time,type=“l”). [↑](#footnote-ref-49)