Análisis de datos

Irene García, Ricardo Alberich, Arnau Mir y Francesc Roselló

2023-09-18

Tabla de contenidos

Presentación

Esto es una edición en línea de los apuntes de Análisis de Datos.

El enfoque es teórico-práctico para el grado de Matemática de la UIB y se puede emplear como un curso previo una asignatura de Análisis de Datos en grados de informática.

Este libro fue editado en con Quarto

1 Introducción al Análisis Multivariante de Datos

1.0.1 La estadística y el método científico

- La ciencia avanza definiendo teorías que intentan explicar el mundo.
- La comunidad científica elabora teorías/hipótesis que intentan explicar hechos que ocurren. Una hipótesis es científica si existe alguna manera de comprobar su veracidad.
- Podemos diseñar experimentos para comprobar si se cumplen las afirmaciones de la teoría.
- Como la naturaleza tiene un comportamiento con "incertidumbre", es decir, que si repetimos el experimento se obtienen resultados similares pero no idénticos, la estadística permite analizar estos resultados y ver si las desviaciones de la teoría son razonables o no.
- Se ha definido estadística de muchas maneras. La que más nos gusta, y que está relaciona con la situación que acabamos de explicar, es que:

  La **estadística** es la ciencia que permite adquirir conocimiento generalizable a partir de datos.

- La estadística ayuda en todas las fases del método científico:
  - *Planteamiento del problema*: Diseño de experimentos y encuestas, determinación del tamaño de la muestra y métodos de muestreo adecuados para garantizar que los datos recopilados sean representativos de la población objetivo.
  - *Recopilación de datos*: Proporciona herramientas para recopilar y organizar datos relevantes sobre el problema.
  - *Análisis de datos*: Aplicación de técnicas descriptivas (Análisis explorartorio de datos), así como técnicas inferenciales (contrastes de hipótesis, ajustes de modelos,etc) para sacar conclusiones sobre la población en función de la muestra recopilada.
  - *Interpretación de resultados*: Ayuda a los científicos a determinar si los resultados son estadísticamente significativos y si las conclusiones se pueden generalizar a la población más amplia.

  - *Comunicación de hallazgos*: La estadística se usa para comunicar los resultados de manera efectiva a través de gráficos, tablas y tests estadísticos. Esto es esencial para que otros investigadores puedan comprender y evaluar los resultados.
  - *Reproducibilidad*: Proporciona métodos estadísticos claros y transparentes, se permite que otros repitan los experimentos y análisis para verificar la validez de los hallazgos.
  - *Toma de decisiones*: En muchos campos científicos, los resultados estadísticos se utilizan para tomar decisiones importantes. Por ejemplo, en la medicina, la estadística se usa para evaluar la eficacia de tratamientos y tomar decisiones sobre su uso en la práctica clínica.

- Cuando alguien realiza un nuevo descubrimiento lo envía a una revisión por pares de la comunidad científica. Para que estos acepten el descubrimiento y pase a formar parte del conocimiento científico debes poner a disposición:
  - Los datos brutos (raw data) junto con el modelo de datos.
  - El código parametrizado y con las líneas más importantes comentadas.
  - La documentación (artículo/ reporte) donde se interpretan y presentan los resultados más relevantes.

En resumen, la estadística es una herramienta esencial que ayuda a garantizar que la investigación científica sea rigurosa, confiable y basada en evidencia sólida.

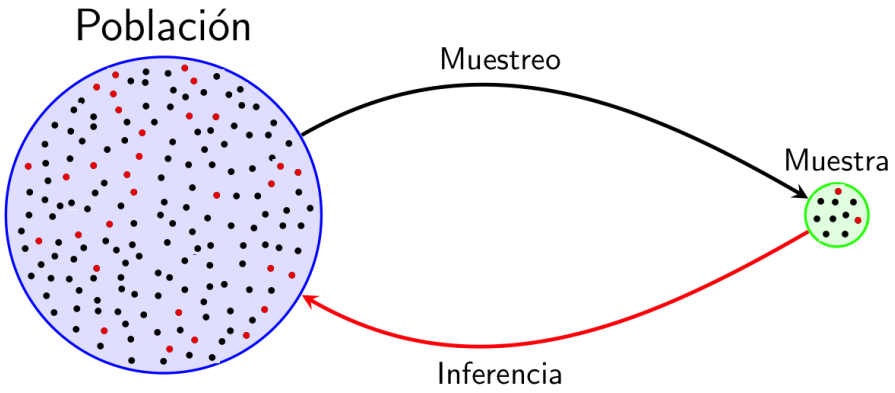
1.1 Gestión básica de datos

1.1.0.1 Introducción

- En estadística, siempre se empieza obteniendo unos **datos** sobre un grupo (relativamente pequeño) de individuos de una población. Bueno, en realidad, no se empieza obteniendo los datos, sino planificando cuidadosamente cómo se van a obtener, pero todo forma parte de la "obtención" de los datos.
- Se **generaliza la información** que se ha obtenido sobre este grupo de personas al total de la población.
- Y no se trata de trucos de magia adivinatoria, sino de una **ciencia** cuya metodología ha sido validada por medio de demostraciones matemáticas o, en el peor de los casos, mediante simulaciones numéricas (el equivalente en matemáticas de los experimentos en las otras ciencias).

Así pues, la situación de partida a la hora de aplicar técnicas estadísticas es que disponemos de un conjunto de datos que describen algunas características de un grupo de individuos. El análisis estadístico de estos datos puede ser entonces de dos tipos básicos:

- **Análisis exploratorio de datos**, cuando nuestro objetivo sea simplemente resumir, representar y explicar los datos concretos de los que disponemos. La **estadística descriptiva** es el conjunto de técnicas que se usan con este fin.
- **Análisis inferencial**, si nuestro objetivo es deducir (**inferir**), a partir de estos datos, información significativa sobre el total de la población de interés. Las técnicas que se usan en este caso forman la **estadística inferencial**.



Ambos tipos de análisis están relacionados. Por un lado, porque es conveniente (obligatorio, en nuestra opinión) empezar cualquier análisis inferencial dando un vistazo a los datos que se usarán.

Por otro, porque muchas técnicas descriptivas permiten estimar propiedades de la población de la que se ha extraído la muestra. Por citar un ejemplo, la media aritmética de las alturas de un grupo de individuos nos da un valor más o menos representativo de sus alturas, pero también sirve para *estimar* la altura media de los individuos de la población total.

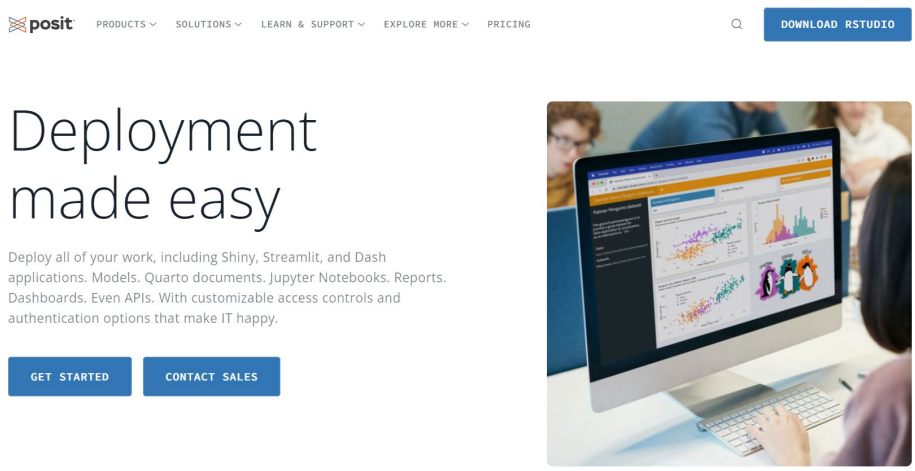
La estadística inferencial entra en juego cuando se quiere obtener información sobre una población y no se puede acceder a todos sus integrantes. Si por ejemplo queremos conocer la altura media de los estudiantes matriculados en esta asignatura de la UIB en este curso, en principio no necesitamos para nada la estadística inferencial. Sois pocos, os mediríamos a todos y calcularíamos la media. En todo caso, usaríamos técnicas de estadística descriptiva para arropar este valor representando la distribución de vuestras alturas de manera adecuada.

Pero si quisiéramos conocer la altura media de los mallorquines entre 18 y 25 años, sería muy complicado medirlos a todos. Entonces, lo que haríamos sería tomar una muestra representativa de esta población, medirlos y a partir de sus alturas estimar dicha altura media. Naturalmente, lo más seguro es que de esta manera no obtuviéramos el valor exacto de la altura media de los mallorquines de 18 años, nos tendríamos que conformar con obtener una aproximación dentro de un cierto margen de error y determinar la probabilidad de acertar con nuestra estimación y este margen de error. La estadística inferencial es la que nos permite acotar el error que podamos haber cometido y calcular la probabilidad de cometerlo, incluyendo la metodología que tendríamos que haber usado para tomar la muestra en primer lugar.

1.1.0.2 R/ RStudio - Posit / RMarkdowm - Quarto

Todas las técnicas que usaremos en la asignatura pueden ser implementadas y/o desarrolladas en software libre como Python y R. Ambos se consideran lenguajes de programación esenciales para la ciencia de datos. Lo ideal sería dominar ambos para tener una base de programación completa, pero:

- R es un **lenguaje específico utilizado para el análisis de datos y la estadística**.
- R es muy adecuado para un sub-campo del aprendizaje automático conocido como aprendizaje estadístico. Cualquier persona con una formación formal en estadística debería reconocer la sintaxis y la construcción de R.
- Al igual que Python, R cuenta con una sólida comunidad, estructurada alrededor de la "Comprehensive R Archive Network", o CRAN, pero no ofrece un desarrollo de software de propósito general como Python.
- Cada día salen nuevos paquetes que extienden las funcionalidades de R y cubren casi todas las necesidades computacionales y estadísticas de un científico. Para que os hagáis una idea, en el momento de revisar estas notas (septiembre de 2023) el número de paquetes en el repositorio de la CRAN acaba de superar los 19800.
- El acceso a R se proporciona a través de RStudio, entorno que presenta una ventana de visualización, un explorador de archivos, un visor de datos y un editor. Este entorno suele ser menos intimidante que el shell de R. Además, cuenta con ayuda integrada, resaltado de sintaxis y completado contextual por tabulaciones; todas estas herramientas facilitan el trabajo.
- RStudio tiene un nuevo nombre desde julio de 2022: **Posit**. Posit es una palabra que significa proponer una idea para su discusión, proviene de la aspiración científica de construir niveles cada vez mayores de conocimiento y comprensión de experimentos que generan



- Posit tiene como misión la creación de software libre y de código abierto para la ciencia de datos, la investigación científica y la comunicación técnica. Han incluido algunas herramientas para Python a través de **Quarto**.
- Quarto está pensado como un cuaderno de laboratorio moderno donde predomina R pero que soporta código Python (`reticulate`), SQL, Julia, entre otros; pensado para experimentos que requieren multilenguaje.
- **Posit Cloud** permite acceder al potente conjunto de herramientas de ciencia de datos de Posit directamente desde su navegador. Esto favorecerá el trabajo en equipo. Podéis revisar la siguiente Guía para crearos una cuenta.

1.1.0.3 Control de versiones con Git / GitHub

- En esta asignatura la forma de llegar a un resultado de análisis de datos es tan importante como el propio resultado. Además, uno de los objetivos es exponeros al uso de herramientas de software para la ciencia de datos moderna.
- La idea de reproducibilidad lleva implícita la colaboración. El código que se produce es parte de la documentación del proceso y es fundamental compartirlo (aunque sólo sea con uno mismo).
- Lo anterior se logra mejor con un sistema de control de versiones distribuido como **Git**. Mantener un registro sobre los proyectos, es lo que permite rastrear y gestionar cambios en el código a lo largo del tiempo. Se puede decir que nos permite guardar el progreso de nuestro código de tal forma que, si en algún momento cometemos algún error irreversible en una versión posterior, siempre podremos recuperar una versión anterior en la que todo funcionaba correctamente y retomar el proyecto desde ese punto.

- Git permite la colaboración, pero carece de características sociales y herramientas específicas para la colaboración en equipo. **GitHub** proporciona herramientas para la revisión de código, la gestión de problemas y la colaboración en proyectos.

- GitHub es un servicio en la nube donde se pueden subir repositorios propios y compartir el código con otras personas de tal forma que sea accesible desde Internet.
- Un repositorio funciona como una carpeta virtual. En él se encuentran todos los archivos de un proyecto y el historial de revisiones de cada uno, permitiendo restablecer una versión del código en caso de error en su ejecución.
- Podemos ver proyectos de otras usuarios, valorarlos, proponer mejoras en el código, GitHub es una de las aplicaciones que mejora la gestión de proyectos y el acceso a recursos compartidos.
- En octubre del 2021 se estrenó **GitHub Copilot**, una herramienta de inteligencia artificial en la nube desarrollada conjuntamente entre GitHub y OpenAI. Su objetivo es sugerir y autocompletar el código escrito en entornos de desarrollo integrados (IDE).

  En esta clase, utilizaremos GitHub como sistema de gestión del aprendizaje para distribuir y recopilar las entregas como repositorios.

- Crearemos en GitHub un repositorio por estudiante/equipo para cada entrega. Utilizaremos un sencillo flujo de trabajo centralizado que sólo requiere realizar acciones simples como push, pull, add, rm, commit, status y clone.

1.1.0.4 Git / GitHub con R

Veamos cómo configurar todo. Gran parte de lo que está aquí proviene del libro Happy Git and GitHub for the useR de Jenny Bryan y del artículo de David Keyes puedes ver sus vídeos en caso de que, la breve explicación que presentamos abajo, no sea suficiente para ti.

- *Instalar Git*: El primer paso es instalar Git, en el Capítulo 6 del libro explican el proceso para los usuarios de Mac, Windows y Linux. Nosotros ya lo tenemos instalado, así que mostramos cómo verificar si tienes Git instalado y su versión usando el terminal en RStudio.

En el terminal de RStudio:

```
which git # ruta donde está instalado el Git
git --version # version
```

- *Configurar Git (Editar gitconfig file)*:El siguiente paso es configurar Git. Esto se trata en el Capítulo 7 del libro, aunque mostramos lo que creemos es un proceso un poco más fácil. Específicamente, sugerimos usar la función `edit_git_config()` del paquete `usethis`, que abrirá su archivo gitconfig. Agrega tu nombre y correo electrónico y cierra esto.

En la consola de RStudio:

```
library(usethis)
usethis::edit_git_config()
# Modificar en el fichero ".gitconfig" los apartados: "name" y "email"
# y guardar el fichero
```

- *Inicializar un repositorio Git*: Ahora que has instalado y configurado Git, puedes usarlo localmente. La función `use_git()` agregará un repositorio Git (a menudo denominado "repositorio") a un proyecto RStudio existente. Aquí crearemos un nuevo proyecto y luego inicializaremos un repositorio de Git.

En RStudio: * Crear un proyecto nuevo

- Seleccionar "Nuevo Directorio"
- Proyecto
  - Activar: "Create a git repository"

En la consola de RStudio:

```
library(usethis)
usethis::use_git()
# Elegir siempre la opción: 1
# Y ante la ventana, seleccionar: "Save"
```

Y visitar la pestaña: "Git" en RStudio.

- *Ver historial de confirmación*: Ahora que tu proyecto de RStudio tiene un repositorio Git asociado, verás una pestaña adicional en la parte superior derecha: la pestaña Git. Desde aquí, puedes ver todo el historial de cambios en tu código a lo largo del tiempo (¡todavía no muchos!).

En RStudio:

Visitar la pestaña: “Git” de RStudio Pulsar el icono del reloj para ver el historial de “Commit” realizados para ver el “Initial Commit”.

- *Hacer una confirmación (commit) y ver más historia*: Git no realiza un seguimiento automático de los cambios de la manera en que lo hace una herramienta como Google Docs. En su lugar, tienes que decirle a Git: Hice cambios y quiero que mantengas un registro de ellos. Decirle a Git esto se llama hacer una confirmación (commit) y puedes hacerlo desde RStudio.

Cada commit tiene un mensaje de confirmación, lo que es útil porque, cuando miras tu historial de código, ves lo que hiciste en cada momento (es decir, en cada commit). RStudio tiene una herramienta integrada para ver su historial de código. Puedes hacer clic en cualquier commit para ver qué cambió, en relación con el commit anterior. Las líneas que se agregaron en verde; y las que se eliminaron en rojo.

En RStudio:

- Crear un fichero de script R: “test.R” y guardarlo.
- Visita la pestaña “Git” de RStudio y pulsa sobre el botón de “commit” para confirmar la creación del fichero: “test.R”.
- En el panel del commit añada un texto que lo defina.
- Haz varios cambios en el fichero “test.R” y en cada uno de ellos haz de nuevo un “commit”.
- Revisa luego la historia de los cambios que se han producido en el historial (pulsar el icono del reloj).
- Observa los nuevos cambios resaltados en color verde. Frente a los valores antiguos que aparecerán en color rojo.

1.1.0.5 Conectar RStudio y GitHub

El proceso hasta ahora nos ha permitido usar Git localmente. Pero, ¿qué pasa si queremos conectarnos a GitHub? ¿Cómo lo hacemos?

La mejor manera de conectar RStudio y GitHub es usando tu nombre de usuario y un token de acceso personal (PAT). Para generar un token de acceso personal, usa la función `create_github_token()` de `usethis`. Esto te llevará a la página correspondiente en el sitio web de GitHub, donde le darás un nombre a tu token y lo copiarás (¡no lo pierdas porque nunca volverá a aparecer!).

En la consola de RStudio:

```
library(usethis)
usethis::create_github_token()
```

- Pulsa sobre el enlace que aparece en la salida en la consola.
- Se abrirá una página web de Github en la que tendrás que pulsar el botón "Generate token".
- Copia el token que aparece en Github (lo utilizarás en el siguiente paso).
- Ahora que has creado un token de acceso personal, debes almacenarlo para que RStudio pueda acceder a él y sepa conectarse a tu cuenta de GitHub. La función `gitcreds_set()` del paquete `gitcreds` te ayudará aquí. Ingresará tu nombre de usuario de GitHub y el token de acceso personal como contraseña (NO tu contraseña de GitHub). Una vez que hayas hecho todo esto, ¡habrás conectado RStudio a GitHub!.

En la consola de RStudio:

```
library(gitcreds)
gitcreds::gitcreds_set()
# Ante la pregunta: "Enter password or token"
# introduce el token copiado en el paso anterior
```

1.1.0.6 Conectar proyectos de RStudio con repositorios de GitHub

Ahora que hemos conectado RStudio y GitHub, discutamos cómo hacer que los dos funcionen juntos. La idea básica es que configures los proyectos que creas en RStudio con repositorios GitHub asociados. Cada proyecto de RStudio vive en un solo repositorio de GitHub.

¿Cómo conectamos un proyecto de RStudio a un repositorio de GitHub? Happy Git and GitHub for the useR propone tres estrategias. Demostraremos la forma más sencilla

crear un repositorio en GitHub primero. Cree el repositorio y, a continuación, cuando inicie un nuevo proyecto en RStudio, utilice la opción de control de versiones, introduzca la URL de su repositorio y listo.

GitHub primero

Crea el repositorio en GitHub y, a continuación, cuando inicies un nuevo proyecto en RStudio, utiliza la opción de control de versiones, introduce la URL de tu repositorio y listo.

Para bajar un repositorio creado en Github a un proyecto local en RStudio, tendréis que realizar los siguientes pasos:

- Crear un nuevo repositorio en nuestra cuenta de Github (o utilizar uno ya existente): pulsar el botón "Create repository".
- Copiar al portapapeles la primera dirección que aparece (pulsando el botón de la derecha). Coincide con la dirección url que aparece en la barra del navegador.
- En RStudio seleccionamos crear "New project", elegimos "Version Control" y luego seleccionamos "Git".
- Introducimos en el primer cuadro de texto la url copiada anteriormente. Pulsamos "Create Project".
- A continuación podrás consultarse la pestaña "Git" y ver la información asociada al repositorio descargado.

1.1.0.7 Flujo de trabajo general

Ahora que hemos conectado RStudio y GitHub, podemos compartir nuestro trabajo entre los dos.

Push (Subir a Github)

"Push" significa enviar cualquier cambio en tu código de RStudio a GitHub. Para hacer esto, primero tenemos que hacer un commit. Después de confirmar, ahora tenemos un botón (la flecha hacia arriba) en RStudio que podemos usar para enviar nuestro código a GitHub.

En RStudio:

- Creamos un nuevo fichero de script R o un fichero Rmd y lo guardamos.
- Pulsamos en la pestaña "Git" sobre el botón de "commit". Marcamos todos los ficheros sobre los checks de "Staged", rellenamos la descripción del commit y pulsamos sobre el botón de "commit".
- Después de hacer el commit, pulsamos sobre el botón "Push" para subir los cambios a Github.
- A continuación puedes comprobar en la página de Github del repositorio que se han actualizado los últimos ficheros considerados en el último commit.

Pull (Descargar desde Github)

Lo opuesto a "empujar"Push" es bajar ("Pull"). Utilizando el botón de flecha hacia abajo, RStudio va al repositorio de GitHub, toma el código más reciente y lo lleva a su editor local.

Hacer "Push" regularmente es extremadamente importante si estás colaborando, aunque si eres el único que trabaja en un proyecto de RStudio y un repositorio GitHub asociado, sabes que tu código local coincide con lo que está en GitHub, por lo que es menos importante.