Automatización con R y Python I

Bloque 3. Tema 6 (6.1. y 6.2.)

unir LA UNIVERSIDAD EN INTERNET

Profesor: Luis F. Ruiz

¿Qué veremos hoy?

01 **El fin del copiar y pegar.**

02 **Informes Dinámicos | Reproducibles.**

03 **Técnicas básicas de modelización predictiva y descriptiva.**

6.1. El fin del "Copiar y Pegar"

El Problema Estratégico

En la analítica tradicional, generar informes consume tiempo valioso y es extremadamente propenso a errores humanos. Cada actualización requiere intervención manual, creando inconsistencias y riesgos en la toma de decisiones.

La Solución: Automatización

Pasamos de documentos estáticos a **documentos reproducibles**. Si los datos cambian, el informe se regenera automáticamente manteniendo formato y coherencia absoluta.

- **Valor para el Negocio:** Reducción de tiempos de entrega y garantía de consistencia en decisiones estratégicas. El objetivo no es solo rapidez, es trazabilidad.

6.2. ¿Qué es un Informe Dinámico y Reproducible?

Un informe dinámico no es un documento de texto normal: es un **guion de ejecución** que integra narrativa con análisis en tiempo real.

01

Lectura del Archivo Fuente

El sistema carga el documento que combina texto narrativo con bloques de código ejecutable.

02

Ejecución en Tiempo Real

Se procesan cálculos estadísticos, análisis de datos y generación automática de gráficos.

03

Compilación Final

Texto y resultados se fusionan en un documento profesional listo para distribución.

Ventaja Clave: Garantiza que la narrativa siempre coincida con los datos. Jamás veréis un texto que diga "las ventas suben" junto a un gráfico que muestra una bajada.

Pipeline de automatización (visión general)

De datos a informe: mismas reglas, nuevos datos, informe nuevo.

Idea clave: el informe se construye desde código y se puede regenerar automáticamente.

Automatización típica: programar ‘render’ (cron/CI) y distribuir (email/repositorio/intranet).

- Tenemos un informe.qmd que dice: “Ventas del mes” + un gráfico hecho con Python/R.
- Cambiamos unos datos (por ejemplo, ventas de 120 a 150).
- Pulsamos **Render** (o ejecutamos quarto render informe.qmd).

➔ Se genera de nuevo informe.html con **la tabla y el gráfico actualizados automáticamente**.

render
cron
CI/CD
commit

RMarkdown: la herramienta clásica

RMarkdown es el **estándar reconocido** para crear documentación viva en el ecosistema R, integrando análisis y narrativa de forma profesional.

Encabezado YAML

Define metadatos esenciales: título, autor, fecha y formato de salida deseado (PDF, HTML, Word).

Cuerpo Narrativo

Texto explicativo en lenguaje natural que contextualiza el análisis y guía al lector.

Code Chunks

Bloques de código R que se ejecutan al "tejer" (*knit*) el informe, generando resultados dinámicos.

chunks
.Rmd
YAML
knit

- **Control de Versiones:** Al ser archivos de texto plano, se integran perfectamente con **Git**, permitiendo auditar quién modificó cada parte del análisis.

Mini-ejemplo RMarkdown (R): ventas por región

````
---
title: "Informe de ventas (ejemplo)"
output: html_document
---

```{r}
library(dplyr)
library(ggplot2)

ventas <- data.frame(
  mes = rep(c("Ene","Feb","Mar"), each = 3),
  region = rep(c("Norte","Centro","Sur"), times = 3),
  importe = c(120, 90, 60, 140, 95, 70, 160, 110, 80)
)

resumen <- ventas %>% group_by(region) %>%
  summarise(total = sum(importe), .groups = "drop")

resumen

ggplot(ventas, aes(mes, importe, color = region, group = region)) +
  geom_line() + geom_point() +
  labs(title = "Ventas mensuales por región", x = "Mes", y = "Importe")
```
````

Salida: tabla de resumen + gráfico; cambia al re-renderizar con nuevos datos.

Quarto: la evolución del estándar

Quarto representa el **salto evolutivo** de RMarkdown, diseñado específicamente para equipos multidisciplinares que trabajan con diversos lenguajes de programación.

Multi-lenguaje Nativo

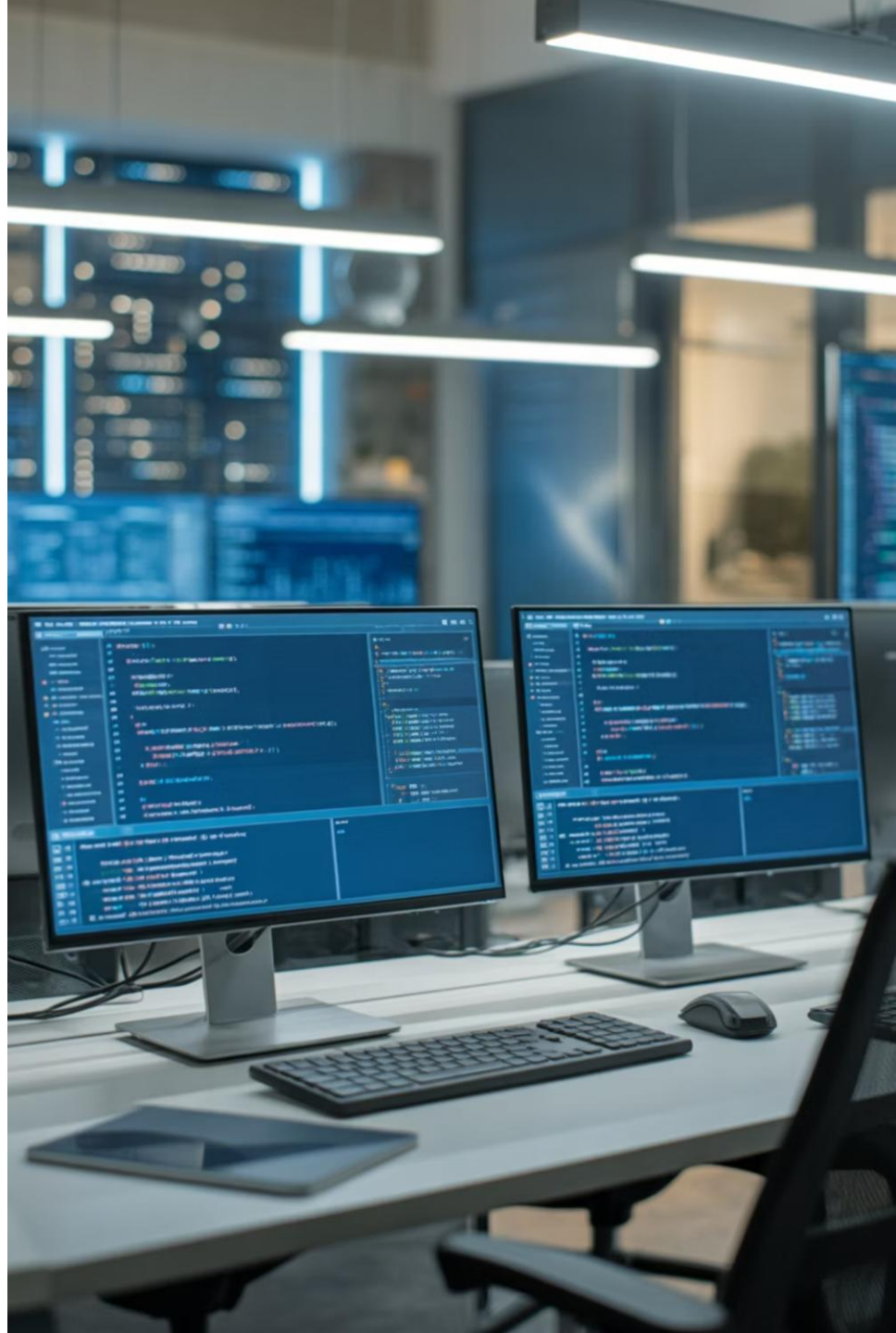
Soporta de forma nativa **R, Python, Julia y Observable** en el mismo documento sin configuraciones complejas.

Flexibilidad Extrema

Permite limpiar datos con **Python (Pandas)** y visualizarlos con **R (ggplot2)** en el mismo informe.

Salida Simultánea

Genera versiones web interactivas (HTML) y documentos estáticos (PDF) desde el mismo origen.



Mini-ejemplo Quarto (Python): ventas por región

````
---
title: "Informe de ventas (ejemplo)"
format: html
---

```{python}
import pandas as pd
import matplotlib.pyplot as plt

ventas = pd.DataFrame({
  "mes":    ["Ene","Ene","Ene","Feb","Feb","Feb","Mar","Mar","Mar"],
  "region": ["Norte","Centro","Sur"] * 3,
  "importe":[120, 90, 60, 140, 95, 70, 160, 110, 80]
})

resumen = ventas.groupby("region", as_index=False)["importe"].sum()
resumen

for region, df_r in ventas.groupby("region"):
  plt.plot(df_r["mes"], df_r["importe"], marker="o", label=region)

plt.title("Ventas mensuales por región")
plt.xlabel("Mes"); plt.ylabel("Importe")
plt.legend()
plt.show()
```
````


Mismo patrón: YAML + chunk ejecutable; Quarto incrusta tabla y gráfico.

Comparativa: RMarkdown vs. Quarto

Característica	RMarkdown	Quarto
Motor Principal	Depende de R (Knitr)	Independiente (Knitr/Jupyter)
Usuarios Ideales	Analistas centrados en R	Equipos mixtos (R + Python)
Formatos de Salida	PDF, HTML, Word	PDF, HTML, Word, ePub, Webs
Uso de Python	Complejidad alta	Nativo y sencillo
Mantenimiento	Estable pero limitado	Desarrollo activo

> **Recomendación Profesional:** Si iniciáis un proyecto hoy, usad **Quarto**. Es el futuro de la documentación científica y técnica, con soporte corporativo de RStudio/Posit.

RMarkdown vs Quarto: decisión práctica

RMarkdown

✓ Muy extendido en proyectos R
✓ Excelente para reporting en R
✓ Ideal si tu stack es 100% R

Quarto

✓ Evolución moderna multilenguaje
✓ Marco común R + Python
✓ Muy cómodo para proyectos y publicación

Regla simple: equipo mixto (R+Python) ➔ Quarto suele facilitar colaboración.

Caso Real: Automatización de Resultados Financieros

Imaginad el informe de resultados trimestrales de vuestra compañía ejecutándose sin intervención humana:

Extracción Automática

Script de **Python** conecta al ERP y extrae datos de facturación actualizados cada noche.

Procesamiento Inteligente

Cálculo automático de márgenes, desviaciones presupuestarias y KPIs críticos del negocio.

Generación con Quarto

Compilación del informe insertando gráficos actualizados, tablas y análisis en formato corporativo.

Distribución Programada

Envío automático del PDF a Dirección cada lunes a las 8:00 AM vía correo electrónico.

Resultado: Cero intervención humana y riesgo de error minimizado. El equipo puede centrarse en análisis estratégico en lugar de tareas repetitivas.

Buenas prácticas, error típico y cierre

Usa rutas relativas y estructura de proyecto clara (evita rutas absolutas).
Controla dependencias: paquetes instalados y versiones coherentes.
Documenta supuestos y transformaciones dentro del informe.
Renderiza de forma repetible: mismo entorno; fija semilla si aplica.
Resumen: dinámico = se actualiza; reproducible = rehacible y fiable.

Error típico: rutas absolutas (ej. C:\Users\...)
Solución: rutas relativas + proyecto bien organizado.

Resumen y Conclusiones Clave

Reproducibilidad Garantizada

Los informes aseguran que los resultados siempre reflejen el estado exacto de los datos y el código utilizado, permitiendo auditorías completas.

Herramientas Profesionales

RMarkdown es excelente para proyectos R puros. **Quarto** es la opción moderna y versátil para equipos que combinan múltiples lenguajes.

Valor Empresarial

La automatización elimina tareas repetitivas, reduce errores humanos y garantiza trazabilidad mediante control de versiones con Git.

Recordad: La transformación digital empieza convirtiendo procesos manuales en flujos automatizados. Estos conocimientos son fundamentales para vuestra carrera profesional en analítica de datos.

Muchas gracias por vuestra atención

www.unir.net