

R Markdown per Presentazioni

Teoria e Fondamenti

Francesca Liguori, Manuel Caccone

11 luglio 2025

Section 1

Introduzione a R Markdown

Cos'è Markdown?

- **Linguaggio di markup utilizzato per creare testo formattato:** come un Word, ma scritto con simboli semplici e non utilizzando i *tag* tipici del linguaggio HTML
- **Semplicità e immediatezza della sintassi:** Si scrive velocemente usando caratteri come `#` per i titoli e `*` per gli elenchi
- **Non necessita di competenze tecniche specifiche:** A differenza di HTML o LaTeX, non serve essere programmatori per usarlo

Benvenuti al Corso

Questo è un **corso introduttivo** sul linguaggio HTML e Markdown.

- “Facile da imparare”
- “Molto usato nel web”
- “[Risorse utili](#)”

Figure 1: Confronto HTML vs Markdown

Esempio - HTML vs Markdown (2/3)

Linguaggio HTML (complesso e tecnico)

```
<h1>Benvenuti al Corso</h1>
<p>Questo è un <strong>corso introduttivo</strong>
  sul linguaggio HTML e Markdown.</p>
<ul>
  <li>Facile da imparare</li>
  <li>Molto usato nel web</li>
  <li><a href="https://developer.mozilla.org/">
    Risorse utili</a></li>
</ul>
```

Esempio - HTML vs Markdown (3/3)

Linguaggio Markdown (semplice e intuitivo)

```
# Benvenuti al Corso
```

Questo è un **corso introduttivo** sul linguaggio HTML e Markdown.

- Facile da imparare
- Molto usato nel web
- *[Risorse utili](https://developer.mozilla.org/)*

Cos'è R Markdown?

- **Sistema di authoring sviluppato da RStudio:** È un'evoluzione di Markdown che permette di includere calcoli e grafici automatici
- **Combina la sintassi di Markdown con codice R eseguibile:** È possibile combinare testo normale con formule e/o grafici che si aggiornano automaticamente modificando i dati di input del codice
- **Consente la creazione di documenti dinamici e riproducibili:** I report creati si rigenerano automaticamente quando cambiano i dati di input
- **Supporta molteplici formati di output:** Dallo stesso file è possibile ottenere PDF, pagine web, presentazioni o documenti Word

Section 2

R Markdown per Presentazioni

- **Slidy**: Presentazioni web standard compatibili con tutti i browser
- **Beamer**: Presentazioni PDF professionali basate su LaTeX, ideali per stampa e distribuzione
- **PowerPoint**: Il classico formato Microsoft, familiare a tutti e modificabile dopo la generazione

YAML Header per presentazioni

L'intestazione YAML è come un “menu di configurazione” che dice a R come generare la presentazione:

```
---  
title: "Titolo della Presentazione"  
author: "Nome Autore"  
date: "Data"  
output:  
  ioslides_presentation: default  
  # oppure  
  slidy_presentation: default  
  # oppure  
  beamer_presentation: default  
  # oppure  
  powerpoint_presentation: default  
---
```

Struttura di una presentazione in R Markdown

- **Intestazione YAML:** La “carta d'identità” del documento che definisce titolo, autore e formato di output
- **Diapositive:** Si creano automaticamente ogni volta che viene scritto un titolo con `#`
- **Sottosezioni:** I sottotitoli con `##` dividono il contenuto all'interno della stessa slide
- **Contenuto:** È possibile inserire testo normale, elenchi puntati, immagini, tabelle, grafici e calcoli in tempo reale

Section 3

Codice R nelle presentazioni

Includere blocchi di codice

- **Blocchi di codice R:** Si scrivono tra ````{r}` e `````, come “scatole” in cui sono effettuate le analisi dei dati di input
- **Opzioni dei chunk:** Controllano se mostrare il codice, eseguirlo, o nascondere i risultati

```
# Questo è un esempio di blocco di codice R  
plot(cars) # Crea un grafico automaticamente
```

Opzioni chiave per i chunk di codice

- **echo**: Decide se mostrare il codice sorgente nel report (TRUE) o nascondere (FALSE)
- **eval**: Stabilisce se eseguire il codice (TRUE) o mostrarlo solo come esempio (FALSE)
- **include**: Controlla se includere i risultati nel documento finale o calcolarli solo in background
- **fig.width, fig.height**: Definiscono le dimensioni dei grafici generati automaticamente
- **cache**: Memorizza i risultati per velocizzare la compilazione quando il documento viene riaperto

- **Pubblicazione online:** Caricate su RPubS o RStudio Connect per condivisione immediata via link
- **Conversione a PDF:** Ideale per la distribuzione offline o per la stampa professionale
- **File HTML autonomi:** Include tutte le immagini e i grafici in un singolo file facile da condividere
- **Hosting web:** Pubblicate gratuitamente su GitHub Pages o altri servizi web

Section 4

PowerPoint con R Markdown

Generare presentazioni PowerPoint

È possibile creare presentazioni PowerPoint partendo da R Markdown usando dei template aziendali:

```
---  
title: "Presentazione PowerPoint con R Markdown"  
output:  
  powerpoint_presentation:  
    reference_doc: modello.pptx  
    # Template aziendale di riferimento  
---
```

- **Familiare per molti utenti:** Tutti sanno usare PowerPoint, quindi nessuna resistenza al cambiamento
- **Utilizzo di modelli personalizzati esistenti:** Mantenere i template aziendali con loghi e colori già approvati
- **Facile modifica post-generazione:** Dopo la creazione, si possono ancora modificare manualmente le slide se necessario
- **Compatibilità con ambienti aziendali:** Funziona perfettamente con Office 365 e sistemi IT aziendali

- **Meno controllo rispetto ad altri formati:** Il layout è meno prevedibile rispetto a formati web o PDF
- **Layout automatico talvolta imperfetto:** R Markdown fa del suo meglio, ma potrebbe richiedere aggiustamenti manuali
- **Funzionalità interattive limitate:** Non potete includere elementi dinamici o interattivi tipici delle presentazioni web

- **HTML Widgets:** Grafici interattivi (plotly, leaflet, etc.) che il pubblico può esplorare cliccando e zoomando
- **Shiny:** Integrazione con applicazioni web interattive usando `runtime: shiny` per creare vere e proprie app

Section 5

Dashboard e presentazioni HTML dinamiche

- **R Markdown per Dashboard:** Non solo presentazioni, ma veri e propri cruscotti aziendali interattivi
- **Comunicazione visiva efficace:** Le dashboard mostrano grandi quantità di dati in modo immediato e comprensibile
- **Pacchetto flexdashboard:** Lo strumento principale per trasformare analisi R in dashboard professionali

- **Conversione automatica:** Trasforma un documento R Markdown normale in una dashboard con layout a griglia
- **Ideale per gruppi di visualizzazioni:** Perfetto per mostrare più grafici e tabelle correlati insieme
- **Responsive design:** Si adatta automaticamente a computer, tablet e smartphone

Link utili per approfondimenti

- Guida completa alle funzionalità
- Galleria di esempi pratici

Una dashboard flexdashboard può includere tutti questi elementi:

- **Visualizzazioni interattive JavaScript:** Grafici che rispondono al click e permettono zoom e filtri
- **Grafici R tradizionali:** Tutti i tipi di grafici R, integrati perfettamente
- **Tabelle dati interattive:** Con funzioni di ordinamento, ricerca e paginazione automatiche
- **Value boxes:** Riquadri colorati per evidenziare KPI e metriche importanti
- **Indicatori gauge:** Strumenti simili a tachimetri per mostrare performance rispetto a target
- **Testi e annotazioni:** Spiegazioni e commenti per guidare l'interpretazione dei dati

La struttura si basa su un sistema gerarchico di intestazioni Markdown:

- **Intestazioni livello 1 (#)**: Creano nuove pagine della dashboard con tab di navigazione separati
- **Intestazioni livello 2 (##)**: Definiscono colonne (o righe se cambiate orientamento) all'interno di ogni pagina
- **Intestazioni livello 3 (###)**: Creano singoli pannelli all'interno di colonne, ognuno con il proprio titolo

È possibile organizzare dashboard complesse in **pagine multiple** usando le intestazioni di livello 1, ciascuna con la propria scheda di navigazione. Il sistema include anche una **barra di navigazione personalizzabile** per un aspetto professionale.

Il pacchetto offre due approcci principali per aggiungere interattività:

HTML Widgets (interattività lato client):

- **Funzionamento:** Tutta l'interattività avviene nel browser, senza bisogno di server
- **Vantaggi:** Distribuite il file HTML e funziona ovunque, anche offline
- **Esempi pratici:** Disponibili al seguente link
- **Limitazione:** Con dataset molto grandi può rallentare il browser

Shiny (interattività lato server):

- **Funzionamento:** L'interattività richiede un server che esegue codice R in tempo reale
- **Configurazione:** Aggiungete runtime: shiny nell'intestazione YAML del documento
- **Vantaggi:** Potete fare qualsiasi calcolo R interattivo, anche complesso
- **Svantaggi:** Richiede un server online per funzionare, non può essere distribuito come semplice file

```
---  
title: "Shiny Web App"  
output: html_document  
runtime: shiny  
---
```

Section 6

Introduzione a Shiny

Shiny: da semplice analisi a servizio web

Shiny trasforma **le analisi R in applicazioni web professionali accessibili a chiunque abbia un browser**:

PRIMA (processo tradizionale):

Analisi in R → Export manuale in Excel → Copia in PowerPoint → Se cambiano i dati, rifare tutto da capo

CON SHINY (processo automatizzato):

Analisi in R → Interfaccia web sempre aggiornata → Chiunque può modificare parametri e vedere risultati istantanei

I due componenti fondamentali

Ogni applicazione Shiny ha una struttura semplice con due parti essenziali:

1. UI (User Interface) - L'interfaccia utente

Definisce **tutto quello che l'utente vede e può toccare**:

- **Controlli di input**: Pulsanti, slider, menu a tendina, caselle di testo per inserire parametri
- **Aree di output**: Spazi dedicati a grafici, tabelle, testi di risultato
- **Layout e design**: L'aspetto generale della pagina, colori, disposizione elementi

2. Server - La logica di calcolo

Definisce **cosa succede dietro le quinte quando l'utente interagisce**:

- **Lettura input**: Raccoglie automaticamente i valori inseriti dall'utente
- **Esecuzione calcoli**: Esegue le vostre formule e analisi R
- **Generazione output**: Produce grafici, tabelle e risultati da mostrare

Concetto chiave:

Quando l'utente modifica qualsiasi input, Shiny **automaticamente e istantaneamente**:

- ❶ ***Rileva il cambiamento***: Monitora continuamente tutti gli input dell'utente
- ❷ ***Riesegue solo il necessario***: Calcola solo quello che è effettivamente cambiato
- ❸ ***Aggiorna solo gli output interessati***: Modifica solo i grafici e tabelle collegati a quell'input

La reattività: il cuore di Shiny (2/2)

Esempio pratico:

- L'utente sposta lo slider dell'età da 30 a 45 anni
- Il calcolo del premio si aggiorna automaticamente
- Il grafico dei rischi si ridisegna istantaneamente
- La tabella delle riserve matematiche si rinnova

Tutto avviene senza cliccare alcun pulsante “Aggiorna” o “Calcola”!

Dashboard HTML con flexdashboard e shiny

Per combinare flexdashboard con Shiny:

Configurazione nell'intestazione YAML:

```
---  
title: "Dashboard Interattivo"  
output:  
  flexdashboard::flex_dashboard:  
runtime: shiny # Questa riga attiva l'interattività  
---
```

Risultato: Dashboard visivamente accattivanti con controlli interattivi completi.

- Sito ufficiale Shiny con tutorial e esempi
- Guida e installazione su R

Deployment: Come Pubblicare le Tue Applicazioni Shiny

Una volta sviluppata la tua applicazione Shiny, il passo successivo è renderla accessibile agli utenti.

Perché è Importante il Deployment?

Le applicazioni Shiny create localmente funzionano solo sul tuo computer. Per condividerle con colleghi, clienti o utilizzarle in produzione, è necessario pubblicarle su un server accessibile via internet o intranet aziendale.

Opzioni di Deployment per l'Ambiente Aziendale

- **RStudio Connect (Soluzione Enterprise):** una piattaforma professionale progettata specificamente per ambienti aziendali
- **Shiny Server Open Source:** una versione gratuita di server Shiny che si può installare sui server aziendali
- **Soluzioni Cloud:** ShinyApps.io (di RStudio), Amazon Web Services (AWS), Microsoft Azure, Google Cloud Platform

Section 7

Vantaggi e sfide delle presentazioni HTML dinamiche

- **Esplorazione real-time:** modificando parametri dal vivo durante la presentazione per analizzare molteplici scenari
- **Comprensione approfondita:** Possibilità di fare drill-down sui dettagli di maggiore interesse
- **Esplorazione personalizzata:** Ogni spettatore può concentrarsi sugli aspetti più rilevanti per il proprio ruolo
- **Standardizzazione dei processi:** Metodologie analitiche uniformi e riproducibili
- **Efficienza operativa:** Aggiornamenti automatici eliminano il lavoro manuale ripetitivo

Sfide e limitazioni da considerare

- **Curva di apprendimento iniziale:** Richiede tempo per padroneggiare Shiny, plotly e tecnologie correlate
- **Performance con big data:** Dataset molto grandi possono rallentare l'interattività
- **Debugging complesso:** Trovare errori in applicazioni interattive è più difficile che in report statici
- **Infrastruttura server:** Richiede RStudio Connect o server dedicato per il deployment aziendale
- **Gestione sicurezza:** Controllo accessi utenti e protezione dati sensibili

Situazioni ideali per presentazioni interattive:

- **Meetings con analisi approfondite:** Quando c'è necessità di esplorare diversi scenari già durante il meeting
- **Submission regolatorie:** Quando trasparenza e riproducibilità sono requisiti obbligatori
- **Business review trimestrali:** Per drill-down su performance con diversi livelli di dettaglio

Quando preferire PowerPoint tradizionale

Situazioni dove PowerPoint resta superiore:

- **Presentazioni executive narrative:** Quando l'obiettivo è raccontare una storia con messaggio chiaro
- **Conferenze e eventi pubblici:** Con audience numerosa e diversificata che non interagisce
- **Quick update operativi:** Per aggiornamenti rapidi senza necessità di analisi approfondite
- **Situazioni offline garantita:** Quando non potete dipendere dalla connessione internet

Section 8

Flusso di lavoro efficace

Best practices per progetti R Markdown

- **Separazione contenuto e stile:** Tenere dati, analisi e formattazione in file separati per facilità di manutenzione
- **Template riutilizzabili:** Creare modelli standard per report ricorrenti, risparmiando tempo e garantendo coerenza
- **Controllo versione:** Usare Git/GitHub per tracciare modifiche e collaborare in team senza conflitti
- **Pre-elaborazione dati:** Script separati per pulizia e preparazione dati prima della generazione report
- **Modularizzazione:** Dividere documenti complessi in sezioni separate (child documents) per gestibilità

Section 9

Conclusioni

Perché utilizzare R Markdown per presentazioni?

Vantaggi trasformativi:

- **Riproducibilità perfetta:** Codice e presentazione convivono eliminando errori di trascrizione e garantendo tracciabilità
- **Efficienza operativa:** I risultati si aggiornano automaticamente quando cambiano i dati di input, senza intervento manuale
- **Versatilità di output:** Dalla stessa fonte è possibile report PDF, presentazioni web, dashboard interattive per diversi pubblici
- **Integrazione seamless:** Combinazione naturale di codice, testo narrativo e visualizzazioni in un workflow unico
- **Interattività:** La possibilità di esplorare dinamicamente i risultati crea valore aggiunto

Section 10

Risorse e approfondimenti

Libri di riferimento completi:

- **“R Markdown: The Definitive Guide”**: La guida completa e autorevole scritta dai creatori di R Markdown
- **“R Markdown Cookbook”**: Ricette pratiche per problemi specifici e casi d'uso avanzati
- **RStudio R Markdown Documentation**: Documentazione ufficiale sempre aggiornata con esempi pratici

Strumenti per dashboard e applicazioni:

- **flexdashboard**: Guida completa con esempi passo-passo
- **plotly for R**: Tutorial interattivi per grafici dinamici professionali
- **Shiny tutorials**: Corso ufficiale dalla teoria alla pratica
- **Gallery di esempi**: Raccolta di applicazioni per ispirazione e apprendimento

Section 11

Grazie per l'attenzione!

- **Demo interattiva:** Esempi di report e dashboard Rmarkdown
- **Dati sintetici e modelli GAN**

- ① Xie, Y., Allaire, J.J., & Golemund, G. (2018). *R Markdown: The Definitive Guide*. Chapman & Hall/CRC.
- ② Allaire, J.J., Xie, Y., McPherson, J., et al. (2021). *rmarkdown: Dynamic Documents for R*.
- ③ RStudio Team (2022). *R Markdown Reference Guide*.
- ④ Xie, Y. (2020). *Dynamic Documents with R and knitr*. 2nd edition. Chapman & Hall/CRC.
- ⑤ Bryan, J., Hester, J., & Robinson, D. (2019). *What They Forgot to Teach You About R*.
- ⑥ Gesmann, M. (2019). *Reproducible Actuarial Analytics using R*.
- ⑦ Chang, W., Cheng, J., Allaire, J.J., et al. (2023). *Shiny: Web Application Framework for R*.
- ⑧ Sievert, C. (2020). *Interactive Web-Based Data Visualization with R, plotly, and shiny*. Chapman & Hall/CRC.