Introduzione

Quando si cita la *data visualization* a una persona che non la conosce, magari aggiungendo che si tratta di rappresentare dei dati e dei risultati di analisi di dati con *figure*, talvolta anche *interattive*, la reazione che si osserva è spesso quella di chi ha di fronte qualcosa che sembra interessante (finalmente, non il solito astruso linguaggio di programmazione e le linee di codice degli informatici), ma che non si sa esattamente in cosa consista. In fondo, se abbiamo una tabella con dei dati e vogliamo produrre un grafico, non basta cercare in un menù, scegliere la figura stilizzata del grafico che desideriamo creare e cliccare? C'è così tanto da dire da riempire un intero libro? Quando poi si aggiunge che questi descritti nel libro sono strumenti grafici del tutto diversi da quelli di office automation e che a dire la verità non ci si ferma nemmeno ai grafici, pure se interattivi, ma ci sono anche le *dashboard*, cioè l'ultima evoluzione della data visualization, quando si creano vere e proprie applicazioni web dinamiche, allora l'espressione dell'interlocutore o interlocutrice generalmente viene attraversata da un'ombra di preoccupazione. In quel momento si può calare l'asso nella manica dicendo che nella data visualization ci sono anche le mappe, le mappe geografiche, certamente, anche quelli sono dati, sono dati spaziali, dati geografici, e si producono le mappe con lo zoom, le bandierine, le aree colorate e anche mappe cartografiche, mappe di Roma, Venezia, New York...

A quel punto l'interlocutore o interlocutrice ha perso i riferimenti che aveva dall'esperienza consueta con gli strumenti comuni e non sa proprio cosa sia questa data visualization, solo che effettivamente pare che ci sia parecchio da dire, tanto da riempire un libro intero. Se qualcuno si riconosce in questo immaginario interlocutore o interlocutrice (immaginario fino a un certo punto, a dire il vero) sappia che è in buona compagnia. Buona in senso letterale non figurato, perché la data visualization è la Cenerentola della data science che in tanti da una certa distanza ammirano, arriva per ultima e sul più bello è costretta a dileguarsi perché non c'è più abbastanza tempo, eppure sono in molti coloro che, con la giusta occasione di studiarla e praticarla, intuiscono che potrebbe essere decisamente interessante, certamente rivelarsi utile e applicabile in un'infinità di ambiti. Questo per una proprietà che la data visualization ha ed è invece assente nell'analisi dei dati o nello sviluppo del codice: *stimola la creatività visiva insieme a quella logica*. Anche statistici e programmatori usano la creatività, chi lo nega non è mai stato nessuno dei due, creatività logica però. Con la data visualization entra in gioco un'altra dimensione della data science altrimenti negletta, *il linguaggio visua-*

le coniugato con la logica computazionale: i dati sono rappresentati con una forma espressiva non più solo logica e simbolica, ma anche percettiva, sensoriale; entrano in gioco forme, colori, uso e proiezioni dello spazio. La data visualization convoglia conoscenze e logiche differenti per una forma espressiva che ha sempre una doppia anima, computazionale per i dati che la alimentano, visuale e talvolta interattiva per il linguaggio che usa per comunicare con l'osservatore. C'è di che riempire non un solo libro.

Organizzazione dell'opera: fondamenti e contenuti avanzati

Il testo è diviso in quattro parti più una quinta di materiale online. La *prima parte* presenta i *fondamenti* della data visualization con Python e R, i due linguaggi e ambienti di riferimento per la data science. Si introducono le tipologie principali di *grafici statici* risultato di una attività di data wrangling (import, trasformazione) e analisi. Le librerie di riferimento per questa prima parte sono *seaborn* per Python e *ggplot2* per R. Sono entrambe librerie grafiche moderne, open source e in costante evoluzione, sia da parte dei *core developer* sia con i contributi delle rispettive *community*, amplissime e vivaci nelle continue innovazioni. *Seaborn* è la più recente delle due e in parte rappresenta un'interfaccia evoluta della tradizionale libreria grafica *matplotlib* di Python, resa più funzionale e arricchita di funzionalità e tipi di grafici diffusi nella moderna data visualization. *Ggplot2* è la tradizionale libreria grafica per R, unanimemente riconosciuta come una delle migliori in assoluto, sia del mondo open source sia delle tecnologie proprietarie. Ggplot è ricchissima di funzionalità di alto livello ed è uno strumento ineludibile per chiunque si approcci alla data visualization.

La *seconda parte* introduce *Altair*, una libreria in ambiente Python in grado di produrre grafici interattivi in formato HTML e JSON, oltre alle versioni statiche in formato bitmap (PNG, JPG) e vettoriale (SVG). Altair è una libreria grafica giovane ma solida perché a tutti gli effetti rappresenta un'interfaccia orientata alla data science di *Vega-Lite*, libreria grafica di affermata tradizione per applicazioni web grazie alla sintassi dichiarativa in formato JSON. Questa seconda parte presenta un livello di difficoltà superiore rispetto alla prima, ma del tutto alla portata di coloro che abbiano acquisito le conoscenze fondamentali date dalla prima parte. Prima e seconda parte coprono circa metà dell'opera.

La *terza* e la *quarta parte* rappresentano *contenuti avanzati* della data visualization. La difficoltà cresce e così l'impegno richiesto, il quale viene però ricompensato perché ci si affaccia su due veri e propri mondi: quello delle *dashboard web* e quello dei *dati spaziali* e delle *mappe geografiche*. Il termine *dashboard* sarà forse nuovo per molti, ma non lo sono le dashboard. Ogni qual volta si accede sul web a degli ambienti che mostrano dei menù e degli oggetti grafici per operare delle scelte e contenuti sotto forma di dati o grafici, quella che si sta usando è con tutta probabilità una dashboard, ovvero un'applicazione web specifica per quel tipo di contenuti con quella organizzazione. Se si utilizzano i dati di ISTAT/Eurostat/OECD, gli Open Data di un ente pubblico, italiano o internazionale, o un'applicazione aziendale interna che visualizza grafici e statistiche, si sta usando una dashboard. Numerosi sistemi e prodotti per realizzare dashboard con tecnologie differenti sono disponibili, è un

mercato vasto e dinamico. Negli ambienti per la data science Python e R esistono due strumenti formidabili, rispettivamente *Plotly/Dash* e *Shiny*. Sono strumenti professionali, l'elenco degli utilizzatori celebri è lungo, ma sono anche strumenti didattici insostituibili per imparare la logica e i meccanismi di base di una dashboard che, essendo un'applicazione web è integrata con la tecnologia tipica delle pagine e dei siti web. Una dashboard Dash o Shiny è però anche altro, è il punto terminale di una *pipeline* iniziata con i fondamenti della data science, import dei dati, data wrangling, data analisi, poi i grafici statici, quelli dinamici e le mappe. La dashboard è il risultato finale nel quale si concentra e si integra tutto, logiche, meccanismi, requisiti e creatività. Tecnicamente sono impegnative per la presenza della logica reattiva che permette di renderle dinamiche e interattive e per l'integrazione di vari componenti. Nel testo sono discussi e sviluppati esempi già di una certa complessità, con soluzioni diverse, dal *web scraping* di contenuti online fino all'integrazione di grafici interattivi Altair.

Il secondo mondo che si apre, quello delle *mappe geografiche*, è innegabilmente affascinante. I *dati spaziali*, le *choropleth map*, le più semplici, quelle colorate, ma anche *mappe cartografiche* e *mappe HTML interattive* sono la declinazione data dalla data science a una disciplina che ha origini antichissime e che tuttora costituisce un ambiente quasi a se stante, la cartografia con mappe ad alta risoluzione e sistemi informativi geografici (GIS), con le sue specializzazioni e competenze. Gli strumenti della data science fino a pochi anni fa non potevano nemmeno sfiorare quel mondo, oggi vi si sono avvicinati in maniera sorprendente. Questo grazie a progressi straordinari nei sistemi e negli strumenti open source, Python ma soprattutto R, che da non disporre di particolari strumenti adatti a supportare dati spaziali, è ora in grado di offrire strumenti formidabili in grado di gestire efficacemente gli *shape file* di una cartografia tecnica e i sistemi di coordinate geografiche secondo gli standard internazionali. Negli esempi presentati sono stati usati file geografici e cartografici da Venezia, Roma e New York con l'obiettivo di mostrare le straordinarie potenzialità offerte dagli strumenti di Python e R.

A chi è rivolta

È semplice specificare in prima battuta a chi si rivolge questo testo: si rivolge a tutti e tutte. Chiunque trovi interessante la data visualization, la immagini utile per il proprio lavoro, studio e per le competenze che si sta costruendo, troverà un percorso di apprendimento che parte dai fondamenti e si spinge fino ai confini della cartografia e delle applicazioni web. Certo, dire "è rivolto a tutti" è semplice, poi possono venire dei dubbi in chi legge: «Ma anche io faccio parte di quei tutti?». Provo a fare un elenco di chi sono quei «tutti». Possono essere studenti, ricercatori e docenti di scienze sociali, politiche ed economiche, i quali, oltre alle visualizzazioni più standard frutto di analisi di dati, possono voler visualizzare dati relativi al movimento di persone e di merci, supply chain globali, logistica, analisi territoriali o etnografiche. Studenti, ricercatori e docenti di marketing, comunicazione, relazioni pubbliche, giornalismo, media e pubblicità, per i quali la rappresentazione interattiva via web e in

generale grafica è una competenza importante e uno strumento necessario. Ancora possono essere interessati studenti, ricercatori e docenti di discipline scientifiche e mediche, spesso a confronto con rappresentazioni grafiche sofisticate, per esempio in biologia o epidemiologia, senza dimenticare che i contributi grafici da parte della community di genomica e biologia molecolare sono tra i più numerosi. Studenti, ricercatori e docenti di ingegneria, gestionale o bioingegneria, per esempio, usano gli strumenti della data science e la visualizzazione ne è parte integrante. Storici, archeologi e paleontologi producono spesso rappresentazioni grafiche di grande qualità, quindi anche a loro il testo può essere utile. L'elenco è già lungo e molti sarebbero ancora da citare.

Pensando ai docenti, questi troveranno parecchi esempi e spiegazioni di aiuto nel presentare i contenuti e organizzare esercitazioni. Ugualmente i professionisti e le aziende, sia per la propria comunicazione aziendale e istituzionale sia per percorsi di formazione professionale.

Quello che cerco di dire è che la data visualization, come la data science tutta, non è una disciplina settoriale per la quale serve avere un background specifico in statistica, informatica, ingegneria o grafica. Tutt'altro, la data visualization, e la data science in generale, è bene che siano il più trasversali possibile, che vengano studiate e utilizzate da tutti coloro che per i loro interessi di studio e di lavoro, nel loro specifico campo, dall'economia alla paleontologia, dalla psicologia alla biologia molecolare, si trovino a lavorare con dei dati, numerici, testuali o spaziali, e che da questi sia utile ricavare rappresentazioni visuali di alta qualità, magari interattive e strutturate.

Che cosa è richiesto e che cosa si impara

Per seguire e apprendere i contenuti del testo è necessario conoscere i fondamenti della data science con Python e R, intendendo questi con le operazioni di import e lettura di dataset e le operazioni di data wrangling tipiche (ordinamenti, aggregazioni, trasformazioni di forma e di tipo, selezioni). Nel testo sono presentati numerosi esempi per i quali si trova sempre anche la parte di data wrangling (i casi dove è più lunga si trovano nel Materiale Online), e per replicare una visualizzazione tutto il codice necessario è disponibile, a partire dalla lettura del dataset Open Data. Per questo non è richiesto di produrre autonomamente la parte preliminare di operazioni sui dati, però è necessario essere in grado di interpretare la logica e le operazioni che vengono eseguite. Da qui la necessità di conoscere i fondamenti, oltre alla possibilità di produrre varianti degli esempi.

Un altro aspetto che potrebbe apparire problematico è la conoscenza dei fondamenti della data science sia con Python sia con R, perché spesso si conosce solo uno dei due ambienti e linguaggi. A questo proposito, mi sento di rassicurare chiunque si ritrovi in questo caso. Se si conoscono le operazioni di data wrangling con R o con Python, interpretare la logica di quelle svolte con l'altro linguaggio richiede poco sforzo, al più saranno i dettagli della sintassi a sfuggire. Ma qui entra in gioco una seconda considerazione: la conoscenza sia di Python sia di R è particolarmente utile nella moderna data science, chi conosce uno dei due ha soltanto bisogno di una buona occasione per imparare il secondo, scoprendo che la curva di ap-

prendimento è parecchio più lieve di quello che poteva immaginare e lo sforzo risulterà più che ragionevole. Il vantaggio invece sarà notevole in termini di funzionalità e nuovi strumenti che diventeranno disponibili.

L'organizzazione in parti suggerisce anche una progressione e una suddivisione nell'apprendimento e nella didattica. La *prima parte* è adatta anche per chi abbia appena appreso i fondamenti della data science e può essere svolta in parallelo con lo studio di quelli. La maggior parte dei grafici statici richiede operazioni di data wrangling di base e il generare grafici può rappresentare un ottimo strumento didattico per dimostrare la logica e l'uso delle operazioni di data wrangling. La presentazione dei diversi tipi di grafici statici segue un ordine di complessità crescente, dai primi intuitivi e facilmente modificabili in infinite varianti, fino agli ultimi che richiedono la conoscenza di alcune importanti proprietà dell'analisi statistica. Il livello di difficoltà della scrittura del codice è generalmente basso. La *seconda parte* è una naturale prosecuzione della prima. La libreria Altair ha una sintassi lineare e chiara, quindi la maggiore difficoltà introdotta dalle funzionalità interattive, soprattutto in termini di logica computazionale, risulta del tutto alla portata per chiunque abbia appreso i fondamenti contenuti nella prima parte. Il risultato sarà motivante, i grafici interattivi Altair sono di ottima qualità, permettono svariate configurazioni e soluzioni alternative.

Tra queste due parti e le successive *terza* e *quarta* esiste una cesura in termini di cosa è richiesto e cosa si impara, per questo motivo nella parte introduttiva iniziale le ultime due parti sono state presentate come *contenuti avanzati*. È necessario aver acquisito una buona familiarità con i fondamenti, nel cercare informazioni nella documentazione di librerie che si sta iniziando a conoscere, e nel saper gestire con pazienza e metodo gli errori. In altre parole, serve aver fatto un buon numero di esercizi con la parte fondamentale. Servirà anche avere già nozioni, oppure cercare il materiale e studiarlo, sui contesti di riferimento delle due parti.

Per la *terza parte sulle dashboard* è necessario avere nozioni di HTML, dei CSS e in generale di come è fatta una pagina web tradizionale. Non sono nozioni difficili, ma potrebbe servire dedicare un certo tempo per acquisirle. Non sono necessarie nozioni più avanzate, per esempio su JavaScript o framework per applicazioni web. Serve avere maturato una certa sicurezza nello scrivere script in Pyhton perché per le dashboard Python è suggerito l'uso di un IDE tradizionale, al posto di Jupyter Lab. Per le dashboard R, invece, il normale RStudio va bene. In entrambi i casi si imparano i meccanismi reattivi di base per gestire l'interattività, è una logica differente da quella tradizionale, richiede pazienza, si faranno molti errori, e sbagliare è necessario. *Esercitarsi serve a quello, a sbagliare, il più possibile*.

Per la *quarta parte sulle mappe* è necessario imparare le nozioni fondamentali sui sistemi di coordinate geografiche, la forma dei dati geografici con la tipica organizzazione in geometrie e le trasformazioni di coordinate spesso necessarie. Gli strumenti che si usano sono in parte noti, *ggplot* per R e *pandas* per Python, ma molti nuovi si incontreranno perché in ogni caso, non solo nel mondo della cartografia ma anche in quello della data science, la logica, i metodi e gli strumenti per usare dati spaziali hanno delle specificità che vanno apprese. Come citato inizialmente, se c'è una certa difficoltà iniziale da superare ed è richiesto entrare nei dettagli della forma dei dati spaziali, l'uso di questi dati e la produzione di mappe geografiche è affascinante, fin dalle prime e semplici *choropleth map*, nome difficile per la ti-

pologia più comune e nota a chiunque: le mappe con le aree colorate secondo una certa grandezza, come la coalizione che ha vinto le elezioni regionali o la percentuale di vaccinazioni per provincia, regione o nazione. Dopo quelle, arriva la vera bellezza di lavorare con dati spaziali e mappe geografiche.

Che cosa è escluso

Come sempre, o quasi, molto rimane escluso dal contenuto di un libro, talvolta semplicemente per la necessità di non superare un certo numero di pagine, qualche volta per pura dimenticanza, spesso per scelta consapevole dell'Autore. Tutti e tre i motivi esistono anche per quest'opera. Per il primo, in buona parte si è sopperito con il Materiale Online, per il secondo mi spiace ma oltre che scusarmi non so che dire visto che si tratta di cose che ho dimenticato, per il terzo invece qualcosa da commentare c'è, se non altro per dare qualche spiegazione dei motivi delle esclusioni per scelta.

La prima esclusione evidente è l'assenza di tecnologie e strumenti proprietari. Per la data visualization esistono moltissime soluzioni e strumenti proprietari, da molto specialistici prodotti da piccole aziende fino a generalisti prodotti dai big player. I produttori di software per la data visualization potranno dire che i loro strumenti sono migliori di quelli che sono presentati in questo libro. In qualche caso potrebbe forse essere vero, quasi sempre è falso e in generale per poter definire uno strumento migliore degli strumenti open source di Python e R sono necessari parecchi distinguo e precisazioni che raramente vengono chiariti. Una delle principali ragioni che si portano è la facilità d'uso delle interfacce grafiche degli strumenti proprietari rispetto alla programmazione a basso livello di quelli open source. Una questione vecchia, usurata e ormai inattuale che lentamente, forse, si inizia a superare. È ovvio che imparare a cliccare delle sequenze di pulsanti e menù o trascinare icone grafiche sia inizialmente più semplice che scrivere codice con un linguaggio di programmazione. La curva di apprendimento iniziale è diversa nei due casi. Il punto però sta in quell'aggettivo, "iniziale". Cosa succede dopo? Qual è lo scopo per cui si apprende l'uso di questi strumenti? Se lo *scopo* è *didattico*, insegnare e apprendere i fondamenti e i contenuti avanzati della data visualization, non c'è praticamente scelta: solo gli ambienti e gli strumenti che esibiscono i dettagli a basso livello sono strumenti didattici. Gli altri semplicemente non lo sono. Sono adatti a corsi di formazione professionale su quel particolare strumento, ma non a insegnamenti o apprendimento di base. Tanto basta per escludere da questo testo ogni strumento proprietario. Va osservato che alcuni tra i più moderni strumenti proprietari (o forse i produttori più accorti) stanno integrando nei loro framework le tecnologie open source di Python e R.

Ci sono poi due esclusioni specifiche e forse sorprendenti tra le tipologie di grafici di base, e non due del genere esotico che in pochissimi usano, al contrario due tra le più diffuse, diffusissime addirittura, una è perfino la tipologia di grafico più diffusa. Gli esclusi sono i *grafici a torta* (*pie chart*) e le *linee di tendenza* (queste quasi del tutto escluse, a dire il vero). I motivi sono diversi. Per i grafici a torta il motivo è semplicemente che *non sono utili* nel senso proprio della data visualization e della data science. L'affermazione sembrerà sorprenden-

te: in che senso i grafici a torta, onnipresenti e usati milioni di volte, non sono utili? Spiego brevemente il motivo, peraltro condiviso da parecchi che si occupano di data visualization. Si produce un grafico per rappresentare in forma visuale delle informazioni contenute in certi dati e tale rappresentazione si basa su almeno due condizioni: 1) che la rappresentazione visuale sia chiara e interpretabile in modo non ambiguo e 2) che con il grafico le informazioni contenute nei dati siano di più semplice comprensione rispetto alla forma tabellare (o almeno di pari difficoltà). I grafici a torta non soddisfano nessuna delle due condizioni. Sono ambigui perché la dimensione relativa delle fette spesso non è chiara e soprattutto rendono più difficile interpretare i dati rispetto all'equivalente tabella. In altre parole, se si presenta la tabella con i valori invece del grafico a torta, chi legge ha un'informazione più facile, chiara e comprensibile. Perché allora si usano i grafici a torta? Si usano perché aggiungono del colore a un testo altrimenti monotono. E se risultano poco comprensibili come si fa? Si aggiungono i valori alle fette, cioè in pratica si riscrive la tabella dei dati. Per questo i grafici a torta non sono utili ai fini della data visualization ma solo a fini cosmetici. Allora perché non si dice lo stesso per i grafici a barre? I grafici a torta non sono che la rappresentazione in coordinate polari dei grafici a barre. Si dice che il diavolo si nasconda nei dettagli, i grafici a barre, per il fatto di essere in coordinate cartesiane, sono intuitivi e rendono più facile (o almeno non più difficile) interpretare i dati rispetto alla corrispondente tabella. È la stessa differenza tra valutare angoli (difficile) e valutare lunghezze lineari (facile). In ogni caso, produrre grafici a torta è semplice, dopo i primi esempi si è già in grado di farlo, se proprio si vuole. Motivazione differente per escludere la linea di tendenza che serve a indicare come tendenzialmente si distribuisce un campione di punti. Lo fa interpolando i punti e minimizzando le distanze. Non ci sarebbe niente di male se non fosse per il significato che spesso le si attribuisce e che costituisce una delle cause di errori più frequenti e grossolani nell'interpretazione dei risultati di un'analisi, errori per altro presenti in pubblicazioni scientifiche, documenti ufficiali, report pubblici. Nel Materiale Online è disponibile una sezione che sintetizza i criteri per una valutazione corretta dal punto di vista statistico. Invece, dal punto di vista della data visualization, le linee di tendenza sono raramente utili, quasi sempre ridondanti e, come si diceva, spesso inducono interpretazioni del tutto errate. Quando siano utili è difficile dirlo. Se i dati mostrano un'evidente tendenza non serve aggiungere una linea per ribadire l'ovvio, se si ricade nelle condizioni per cui l'approssimazione lineare non è giustificata, si induce l'errore di interpretazione. Rimangono pochi casi e pertanto, in generale, l'uso della linea di tendenza è da sconsigliare. Ne vedremo pochissime.

Per concludere, la data visualization merita più spazio nei programmi didattici e un riconoscimento più chiaro come disciplina e insieme di conoscenze coerente e in evoluzione. Il ruolo di Cenerentola della data science può essere superato, riconoscendone il valore didattico, la valenza formativa e, non meno importante, lo stimolo creativo.

L'opera è arricchita da **Materiali Online**: tutti i dataset usati nel libro,
le figure a colori e le versioni HTML delle figure dinamiche. I materiali sono disponibili all'indirizzo http://mybook.egeaonline.it.