Capitolo 7

Informatica

Nel mondo contemporaneo, l'informatica svolge un ruolo cruciale nell'analisi e nell'interpretazione dei dati provenienti da una vasta gamma di fonti. In questo capitolo, esploreremo due approcci nell'ambito dell'analisi dei dati: l'utilizzo dei fogli di calcolo e la programmazione in Python.

Il COVID-19 ha rappresentato una sfida senza precedenti, richiedendo una valutazione attenta dei dati per comprendere l'andamento della pandemia. Utilizzeremo un foglio di calcolo per analizzare l'andamento della pandemia in Italia, esplorando varie metriche come il tasso di crescita dei casi, il totale degli ospedalizzati e la necessità di utilizzare dei modelli matematici per trarre conclusioni sui dati.

Inoltre, affronteremo la sfida del cambiamento climatico, una delle più urgenti che il nostro pianeta affronta oggi. Utilizzando Python, un potente linguaggio di programmazione per l'analisi dei dati, esamineremo le variazioni delle temperature globali nel corso degli ultimi 60 anni. Utilizzeremo tecniche di analisi dei dati e visualizzazione per identificare trend, anomalie e relazioni significative nel dataset delle temperature globali.

Attraverso queste sezioni, introdurremo lo studente all'importante campo dell'analisi dei dati, utilizzando strumenti pratici e metodologie statistiche per comprendere fenomeni complessi e cruciali per il nostro mondo contemporaneo.

7.1 Foglio elettronico

Un foglio elettronico è un'applicazione software che consente di gestire efficacemente una vasta gamma di dati attraverso calcoli, funzioni matematiche, macro e la creazione di grafici correlati.

Alcuni esempi di fogli elettronici attualmente disponibili sono (Fig.7.1):

- Microsoft Excel;
- Libre Office;
- Open Office;
- Google Sheet.

(a) Microsoft Excel

(b) Libre Office

(c) Open Office

(d) Google Sheet

Figura 7.1: Esempi di fogli elettronici

Nel seguito utilizzeremo il software Google Sheet, ma le operazioni descritte possono essere svolte analogamente con qualunque foglio elettronico.

L'ambiente di lavoro

L'interfaccia grafica di tali software è suddivisa in due parti distinte (Fig. 7.2):

- il *foglio di calcolo*, che è la tabella nella quale vengono scritti i dati;
- l'*ambiente di lavoro*, che è l'insieme delle zone che circondano il foglio di calcolo.

Nell'ambiente di lavoro possiamo distinguere le seguenti parti:

- Barra dei menù: è composta da icone che attivano differenti menù a tendina mostrando le operazioni eseguibili dall'utente;
- Barra degli strumenti: contiene simboli che possono essere utilizzati come "bottoni" per un rapido accesso ad alcune funzioni;
- Barra delle formule: mostra il contenuto della cella selezionata;
- Schede fogli: permettono una migliore organizzazione, attivando fogli di calcolo differenti.

Nel foglio di calcolo invece possiamo definire i seguenti elementi:

- Cella: rappresenta l'unità fondamentale del foglio di calcolo;
- Intervallo: un insieme di più celle;
- Colonna: un insieme verticale di celle;
- Riga: un insieme orizzontale di celle.

Ogni cella può contenere quattro tipologie di informazioni differenti: valore, testo, formula o funzione.

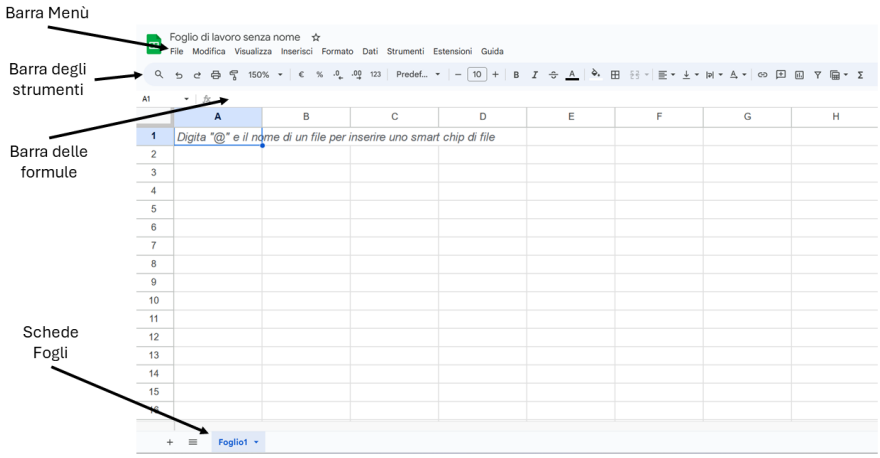


Figura 7.2: Foglio di lavoro

Tipi di dati

In informatica, il *tipo di dato* è un concetto fondamentale che indica l'insieme di valori che una variabile [1] può assumere e le operazioni che possono essere eseguite su di essa. I tipi di dati forniscono una struttura per memorizzare e manipolare dati all'interno di un programma. Essi definiscono le caratteristiche dei dati, come il range di valori che possono essere rappresentati, il modo in cui vengono memorizzati in memoria, e le operazioni che possono essere eseguite su di essi.

Alcuni esempi di formati disponibili nella maggior parte dei fogli elettronici sono rappresentati in tabella (7.1).

Per inserire dati all'interno del foglio di calcolo è sufficiente selezionare la cella desiderata e digitare il contenuto. Un aspetto cruciale all'interno di un foglio di calcolo riguarda i riferimenti, che possono essere distinti in:

[1] Una variabile, nell'ambito dell'informatica, è un'entità che identifica e contiene dati memorizzati in una specifica area di memoria, che può consistere in una o più locazioni di memoria, destinate a contenere valori.

Formato	Descrizione formato	Esempio
Numero	Rappresenta numeri	123,45
Percentuale	Rappresenta percentuali	50%
Valuta	Rappresenta valute	€100
Testo normale	Testo senza formattazione speciale	Ciao
Data	Rappresenta date	06/09/1993

Tabella 7.1: Formati di dati disponibili.

- Riferimenti assoluti, che mantengono la loro posizione invariata durante operazioni di copia;

- Riferimenti relativi, che variano la loro posizione durante operazioni di copia.

Per rendere un riferimento assoluto, si premette il simbolo del dollaro ($) al riferimento della riga o della colonna interessata.

Particolare attenzione deve essere posta nell'inserimento di formule, in quanto *tutte le formule devono iniziare con il simbolo di uguale* "=" (Fig.(7.3)).

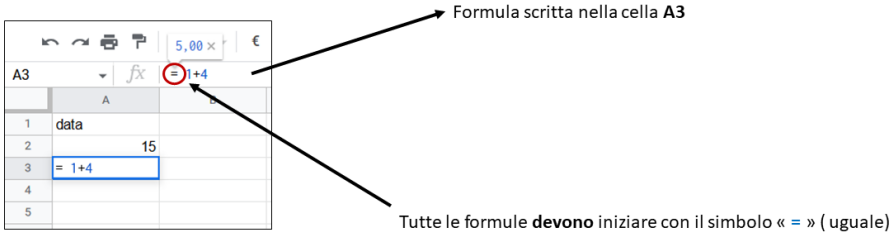
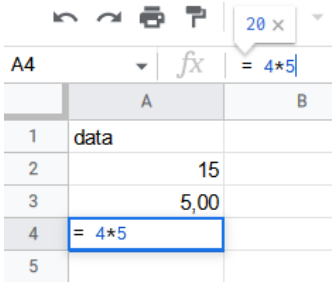


Figura 7.3: Inserimento di dati.

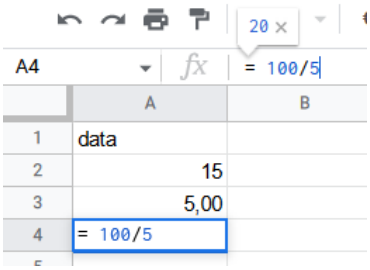
Tutte le operazioni matematiche di base sono supportate dai fogli elettronici: alcuni esempi di operazioni sono riportati in Fig.(7.4).

Operazione	Descrizione
+	Somma
−	Sottrazione
*	Moltiplicazione
/	Divisione
^	Elevamento a potenza

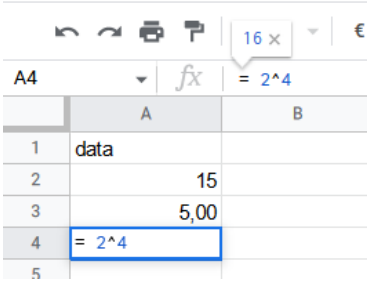
Tabella 7.2: Operazioni matematiche di base.



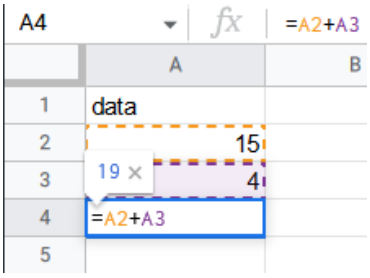
(a) Moltiplicazione



(b) Divisione



(c) Elevamento a potenza



(d) Somma

Figura 7.4: Esempi di operazioni matematiche in un foglio di calcolo.

Le funzioni

Le funzioni rappresentano un elemento essenziale per effettuare calcoli più complessi. Nelle situazioni più basilari e comuni, le funzioni eseguono una serie di operazioni elementari, mentre in circostanze

più complesse svolgono compiti che vanno oltre le operazioni di base, come ad esempio la ricerca nelle celle o il confronto tra di esse.

Per accedere alla lista delle funzioni disponibili, dalla barra menù è sufficiente aprire il menù a tendina "Inserisci" e selezionare la voce "Funzione", come rappresentato in figura (Fig.7.5).

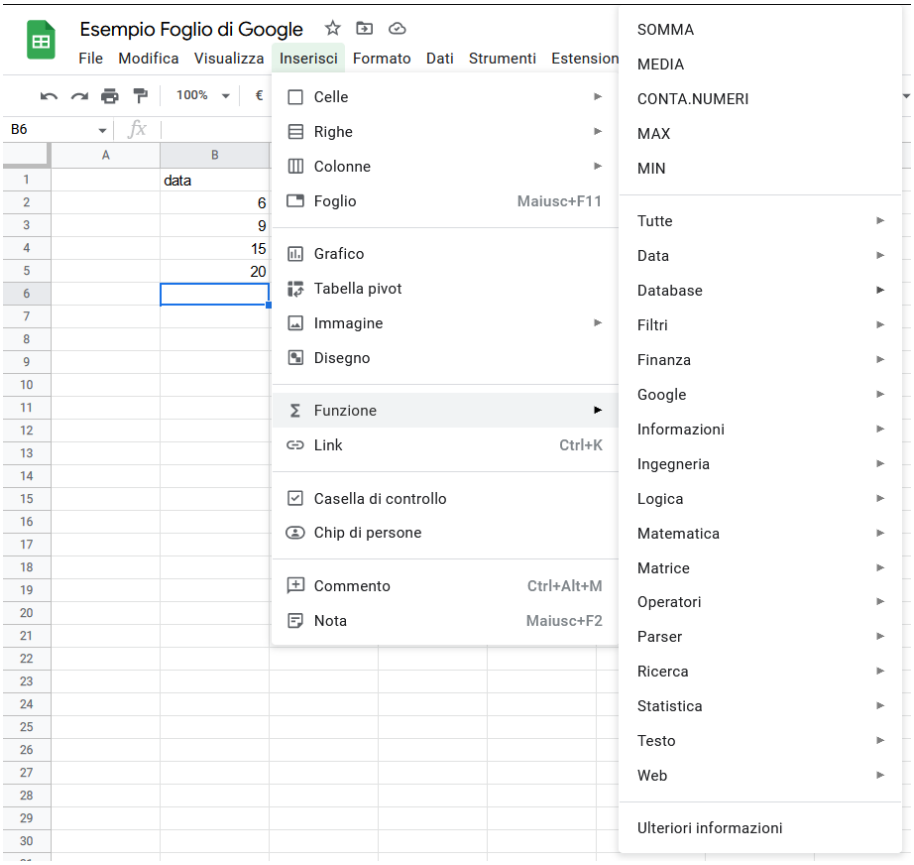


Figura 7.5: Inserimento di funzioni.

Ogni funzione, per essere avviata, richiede l'utilizzo del simbolo di uguaglianza (=), ha un nome che la identifica e può prendere una serie di argomenti o parametri (racchiusi tra parentesi) su cui agisce. Nel caso in cui una funzione richieda più argomenti, essi

devono essere separati da un punto e virgola (;). In Fig.(7.6) è riportato un esempio del calcolo della funzione “media” utilizzando il foglio elettronico.

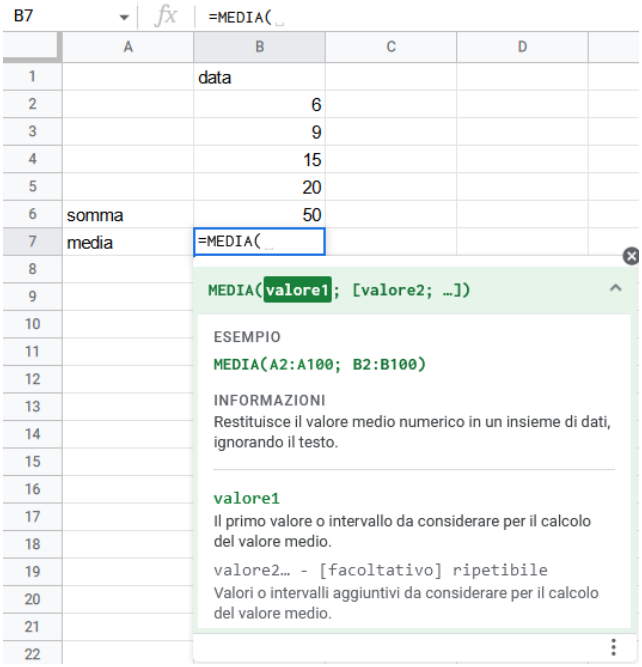


Figura 7.6: Esempio di funzione.

I grafici

Un grafico è una rappresentazione schematica di dati matematico-statistici che permette una più facile e immediata lettura del fenomeno che si sta analizzando.

Con i fogli elettronici si possono fare diverse tipologie di grafici, i più comuni dei quali sono:

- Istogramma: formato da colonne verticali di altezza proporzionale alla quantità rappresentata, risulta utile per confrontare quantità differenti dello stesso fenomeno;

- Grafico a dispersione: due variabili di un insieme di dati vengono rappresentate in uno spazio cartesiano;

- Grafico a linee: serve per rappresentare l'andamento di un fenomeno;

- Grafico a torta: serve a rappresentare un intero fenomeno diviso in parti percentuali.

In un grafico è possibile rappresentare l'andamento di una funzione matematica. Consideriamo per esempio la funzione esponenziale e i valori rappresentati in tabella 7.3.

Funzione esponenziale	
x	$f(x) = \exp\{x\}$
1	2,718
2	7,389
3	20,086
4	54,598

Tabella 7.3: Funzione esponenziale.

Dopo aver riportato i valori all'interno del foglio di lavoro, per poter tracciare il grafico è sufficiente selezionare le due colonne corrispondenti, aprire dalla barra menù la voce "Inserisci" e selezionare "Grafico" (confronta Fig. 7.7).

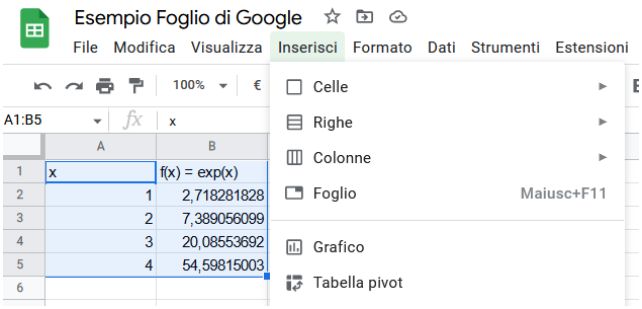


Figura 7.7: Inserire un grafico.

Una volta selezionata la tipologia di grafico è possibile effettuare numerose operazioni utilizzando il menù che si apre nel riquadro dello schermo (per esempio modificare le etichette degli assi, aggiungere serie di dati, ecc.).

Strumento estremamente utile per studiare le funzioni matematiche è la *linea di tendenza*, ovvero una rappresentazione grafica di una serie di dati che cerca di mostrare la direzione generale o la tendenza dei valori. Alcune tipologie di linee di tendenza che possono essere scelte sono

- lineare: la serie di dati viene approssimata mediante una retta $y = mx + q$;
- esponenziale: la serie di dati viene approssimata mediante un esponenziale $y = \alpha \cdot \exp\{\beta x\}$;
- polinomiale: la serie di dati viene approssimata mediante una curva polinomiale $P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_2 x^2 + a_1 x + a_0$.

7.2 Esercitazione con un foglio elettronico

Proponiamo ora un'esercitazione, che consiste nell'utilizzo di un foglio elettronico (Excel, Fogli Google, ecc.) per analizzare dal punto di vista statistico-matematico l'epidemia di Covid-19 che ha colpito l'Italia nel 2019, tracciando grafici e traendo conclusioni dai risultati ottenuti. Nella sezione successiva a questa proporremo un esempio di relazione che risponde a una selezione delle consegne proposte.

Consegna 1: Dataset

Per scaricare i dati necessari all'analisi proposta, è necessario collegarsi al sito del Ministero della Salute:

https://github.com/pcm-dpc/COVID-19

e scaricare i dati contenuti nel file

/dpc-covid19-ita-andamento-nazionale.csv

Si tratta di un dataset in cui vengono raccolte giornalmente diverse informazioni relative all'andamento dell'epidemia di Covid-19 in Italia a partire dal 2020.

Usando i dati a disposizione, è possibile trarre alcune informazioni di carattere generale mediante le funzioni del foglio elettronico.

Consideriamo in particolare la colonna "totale_positivi". Si richiede, usando un foglio di calcolo, di:

1. rappresentare in un grafico (es. Fig. 7.8) l'andamento della pandemia;

2. calcolare tramite le funzioni del foglio di calcolo quanti individui sono risultati positivi dall'inizio della pandemia complessivamente;

3. calcolare quanti massimi locali presenta la funzione ottenuta negli anni 2020-2021.

Figura 7.8: Andamendo della pandemia tra febbraio 2020 e ottobre 2023.

Consegna 2: Approssimazione malthussiana

Consideriamo la prima ondata della pandemia (Febbraio 2020-marzo 2020) e prendiamo in considerazione le prime tre settimane di dati separatamente (24 febbraio - 01 marzo, 02 marzo - 08 marzo, 09 marzo - 15 marzo).

In questi intervalli di tempo, possiamo modellizzare l'andamento dei dati con una curva malthussiana. Per vedere questa corrispondenza seguiamo le seguenti istruzioni per ciascuna settimana considerata.

Si chiede di:

1. Riportare in un grafico la colonna "totale_casi" in funzione del tempo.

2. Utilizzando la funzione "linea di tendenza", modellare i dati con un andamento esponenziale del tipo:

$$n(t) = a * \exp(b * t) \tag{7.1}$$

   dove $n(t)$ è il numero totale dei casi e t è il tempo espresso in giorni [2].

3. Usando il foglio di calcolo, ricavare le equazioni delle curve ottenute selezionando dal menù a tendina "Etichetta" e l'opzione "Utilizza equazione".

Consegna 3: Coefficiente di determinazione R^2

Un'altra funzione che può essere attivata all'interno dei fogli elettronici è il bottone "Mostra R^2".

Il coefficiente di determinazione R^2 ($R^2 \in [0, 1]$) rappresenta un indice statistico che esprime quanto il modello matematico sia accurato per rappresentare i dati: tanto più è vicino a 1, migliore è il modello.

Si chiede di rispondere alle seguenti domande.

[2] Nota: Considerare il primo giorno (24 febbraio) come il tempo $t = 0$.

1. Qual è la curva del tipo (7.1) che meglio rappresenta questa popolazione di dati (Ovvero la curva per cui il coefficiente di determinazione è massimo) [3]?

2. Ricavare i tempi di raddoppio nelle tre settimane prese in considerazione e riportarle in una tabella del tipo seguente.

Risultati forma esponenziale			
Settimane	a	b	t_{double}
I° settimana	226	0.333	$\frac{\ln 2}{b} = 2.08$
II° settimana			
III° settimana			

Consegna 4: Rappresentazione logaritmica

Una stessa popolazione di dati può essere rappresentata nel piano cartesiano utilizzando scale differenti per gli assi.

Ci proponiamo di rappresentare le tre curve precedentemente ottenute usando una scala logaritmica.

L'equazione malthussiana da cui si parte è del tipo:

$$n(t) = a \cdot \exp(bt) \tag{7.2}$$

Nel caso in cui volesse applicare la funzione "logaritmo" a questa equazione, perché l'equazione rimanga valida si deve prendere il logaritmo di entrambi i membri (e non solo del membro di sinistra).

Si può scegliere qualunque base per il logaritmo, ma in questo caso, dal momento che abbiamo già una funzione esponenziale, la scelta più logica è usare il logaritmo in base e (numero di Nepero).

Estraendo quindi il logaritmo dell'eq. (7.2) si ottiene:

$$\begin{aligned} \ln n(t) &= \ln\Big(a \cdot \exp(bt)\Big) = \\ &= \ln a + \ln \exp(bt) = \\ &= (\ln a) + bt \end{aligned}$$

[3] **Non** si deve considerare la curva di tipo "polinomiale".

L'equazione che otteniamo è quindi:

$$\ln n(t) = b \cdot t + \ln a \tag{7.3}$$

che è nella forma

$$Y = m \cdot x + q$$

in cui riconosciamo:

$$\begin{aligned} Y &= \ln n(t) \\ m &= b \\ q &= \ln a \end{aligned}$$

Pertanto la forma analitica da rappresentare è data dall'equazione (7.3).

Si chiede di:

1. A partire dall'equazione precedente, calcolare nuovamente i tempi di raddoppio della popolazione, rappresentando i dati in una tabella come la seguente (vedi Tab. 7.4).

Risultati forma lineare			
Settimane	m	q	t_{double}
I° settimana			
II° settimana			
III° settimana			

Tabella 7.4: Tempi di raddoppio.

2. Rispondere alla domanda: i tempi di raddoppio trovati sono gli stessi trovati precedentemente? Giustificare la risposta.

Consegna 5: L'evoluzione di una popolazione

In questa sezione cerchiamo di capire come effettuare una previsione del valore dei dati, sulla base di un modello di crescita esponenziale.

Consideriamo i dati dal 16 dicembre 2021 [4] al 29 dicembre 2021.

Si chiede di:

[4] Questa data corrisponderà al nuovo $t = 0$.

1. al fine di ottenere un andamento esponenziale quanto più accurato, creare una nuova colonna in cui verrà calcolata la differenza tra la colonna dei "totale_positivi" e il valore costante 305653 [5]

   ("totale_positivi_offset");

2. fare un grafico dell'intero periodo dei "totale_positivi_offset" in funzione del tempo discretizzato $t \in \{0, 1, 2, ..\}$;

3. assumendo un andamento esponenziale dei dati, stimare quanti individui erano positivi il primo gennaio 2022 considerando l'espressione trovata per la funzione e confrontarla con il valore effettivamente registrato.

Consegna 6: Generazione di una Gaussiana

La curva di Gauss [6], conosciuta anche come *distribuzione normale* o *distribuzione gaussiana*, è uno dei concetti fondamentali nella teoria della probabilità e statistica.

[5] Questo valore serve a rimuovere l'offset alla curva e rappresenta il totale dei positivi il 15 dicembre.

[6] **Carl Friedrich Gauss** (1777-1855) è stato uno dei più grandi matematici della storia. Tra i suoi contributi più significativi vi è il teorema fondamentale dell'algebra, che stabilisce che ogni polinomio non costante con coefficienti complessi ha almeno una radice complessa. Ha anche lavorato sulla teoria dei numeri, introducendo concetti come la congruenza e la legge di reciprocità quadratica.

Nel campo della geometria, Gauss ha contribuito a sviluppare il concetto di geometria non euclidea, aprendo la strada allo sviluppo della geometria differenziale e della teoria della relatività di Einstein. Inoltre, ha dimostrato risultati fondamentali per il successivo studio delle curve ellittiche e la teoria delle superfici.

Gauss è stato anche uno dei pionieri nello sviluppo della statistica, definendo la distribuzione normale e il metodo dei minimi quadrati, ampiamente utilizzato nell'analisi dei dati.

La sua genialità e i suoi contributi hanno reso Gauss una figura fondamentale nella storia della matematica e della scienza, e il suo lavoro continua a essere studiato ed ammirato oggi.

La curva di Gauss è caratterizzata da una forma a campana. La sua equazione è data da:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

dove:

- μ, detto *valore atteso*, rappresenta intuitivamente la media della distribuzione;
- σ è la *deviazione standard*, che determina intuitivamente la "larghezza" della curva, cioè la sua dispersione rispetto al valore atteso;
- e è il numero di Nepero.

La curva di Gauss è un modello matematico estremamente importante poiché molti fenomeni naturali e sociali mostrano una distribuzione ben approssimata da essa. Ad esempio, le altezze delle persone in una popolazione, i risultati di misurazioni scientifiche, le valutazioni dei test standardizzati e molti altri dati empirici possono essere approssimati utilizzando la distribuzione normale. È quindi ampiamente utilizzata in diverse aree, quali l'analisi statistica, l'ingegneria, le scienze naturali, le scienze sociali e molti altri campi. La sua comprensione fornisce un fondamento essenziale per l'interpretazione e l'analisi dei dati, nonché per la formulazione di modelli predittivi e la risoluzione di problemi pratici nelle diverse discipline scientifiche e applicate.

Si chiede di:

1. utilizzando la definizione di *distribuzione normale* o *gaussiana* a partire da un insieme di numeri x con cardinalità uguale a 1000 e distribuiti in maniera casuale [7], tracciare il grafico della curva a campana usando un *grafico a dispersione*;
2. usando l'equazione così ottenuta, calcolare la media e la deviazione standard della curva.

[7] Per generare numeri casuali usare la funzione *CASUALE()*.

7.3 Relazione sull'esercitazione con il foglio elettronico

In questa sezione riportiamo un esempio di relazione sullo svolgimento dell'esercitazione della sezione precedente rispondendo alle domande poste in alcune delle consegne.

Introduzione

L'esercitazione condotta consiste nell'analizzare i dati dei casi di Covid-19 in Italia dal 24 febbraio 2020 all'11 ottobre 2023, prelevati dal sito del Ministero della Salute.

Utilizzando il file dpc-covid19-ita-andamento-nazionale.csv, si è eseguita un'analisi mediante fogli di calcolo al fine di valutare l'andamento della pandemia nel periodo considerato e confrontare tali dati con le restrizioni governative, cercando correlazioni tra l'evoluzione della pandemia e le decisioni adottate dal governo, consultate dagli elenchi dei DPCM (Decreto del Presidente del Consiglio dei Ministri).

Il foglio informatico è stato scelto per le sue funzioni automatiche di analisi statistica e creazione di grafici, che hanno agevolato l'analisi dei dati considerati, permettendo di evidenziare i picchi ed eseguire operazioni matematiche.

L'esercitazione si articola in fasi distinte. La prima fase ha coinvolto l'analisi complessiva dei dati dal 24/02/2020 all'11/10/2023, focalizzandosi sul numero totale di positivi. Successivamente sono state esaminate la "prima ondata" (febbraio-marzo 2020) attraverso un'analisi esponenziale dei contagi e la "terza ondata" (16 dicembre 2021 – 29 dicembre 2021) per stimare accuratamente i positivi al primo gennaio 2022.

Considerazioni generali

In questa sezione verranno analizzati i dati relativi ai totali positivi, individui dimessi e guariti, e nuovi positivi dal 24 febbraio 2020 all'11 ottobre 2023.

È stato rappresentato in Fig. 7.9 l'andamento della pandemia durante il periodo considerato. Il picco di positivi è stato registrato tra dicembre 2021 e agosto 2022, con aumenti significativi durante le vacanze natalizie e l'inizio della stagione estiva. Un rapido incremento di casi è stato evidenziato intorno al giorno 240 (20 ottobre 2020), coincidente con il DPCM del 25 ottobre 2020, che ha introdotto ulteriori restrizioni. Inoltre, utilizzando la funzione "somma" del foglio elettronico, è stato calcolato che complessivamente 25.865.855 persone hanno contratto il Covid-19 dall'inizio della pandemia. Si evidenzia il giorno del picco di ospedalizzati, riscontrato al giorno 273 (23 novembre 2020), coincidente con il DPCM del 3 novembre 2020 e l'ordinanza del 27 novembre 2020 che ha confermato le restrizioni precedenti.

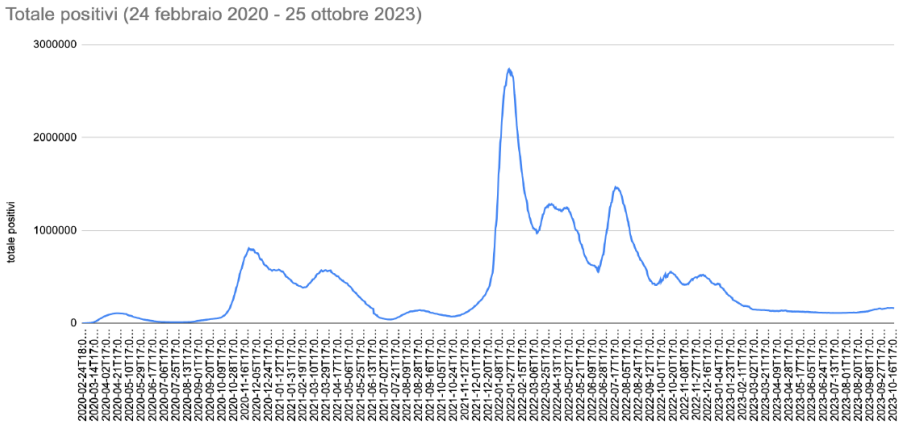


Figura 7.9: Andamento dei dati dal 24 febbraio 2020 all'11 ottobre 2023.

Prima ondata

In questa sezione, vengono esaminati i dati della prima settimana della prima ondata.

Il grafico, arricchito con una linea di tendenza esponenziale del tipo $n(t) = a \cdot e^{bt}$, rivela un notevole andamento esponenziale dei dati (Fig. 7.10), indicando un incremento crescente dei casi di Covid-

Risultati forma esponenziale			
Settimane	a	b	t_{double}
I° settimana	162.17	0.332	2.08

Tabella 7.5: Tempi di raddoppio.

19. È stato scelto questo modello esponenziale poiché il coefficiente di determinazione R^2 ($R^2 \in [0,1]$) si avvicina maggiormente a 1, indicando che il modello matematico è più accurato per descrivere questi dati. Utilizzando la funzione "linea di tendenza" del foglio di calcolo, è stato possibile calcolare il tempo di raddoppio. Poiché il modello esponenziale segue una legge di Malthus con $N_0 = a$, $t = x$ e $\lambda = b$, il tempo di raddoppio t_2 è calcolato come $t_2 = \frac{\ln 2}{b}$. I valori di a, b e t_2 sono riportati nella Tabella 7.5.

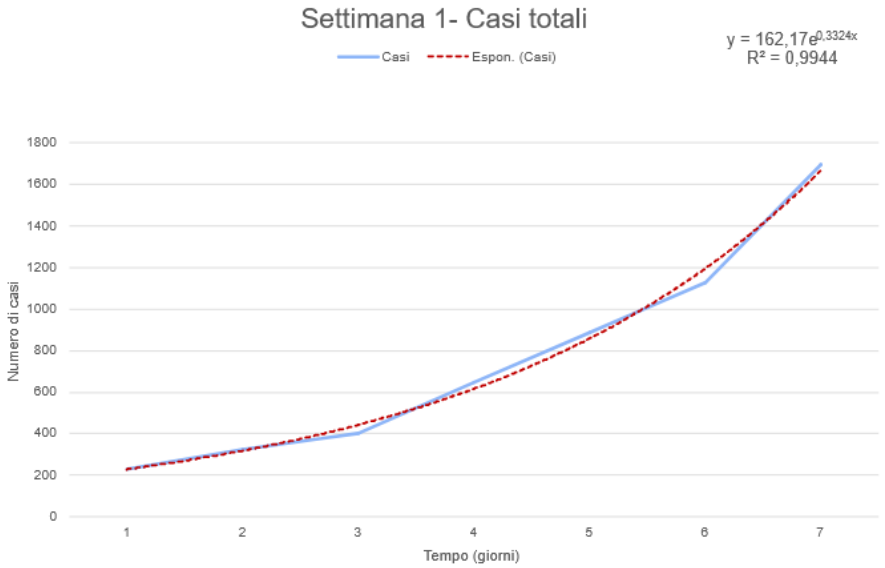


Figura 7.10: Modellizzazione malthussiana dei dati

Nel grafico in figura (Fig. 7.11) vengono rappresentati i risultati in scala logaritmica dei dati già presentati.

Una volta trovata l'equazione $\ln[n(t)] = b \cdot t + \ln(a)$, che è nella forma lineare $y = mx + q$ dove $y = \ln[n(t)]$, $m = b$

Risultati forma lineare			
Settimane	m	q	t_2
I° settimana	0.3324	4.7562	2.08

Tabella 7.6: Tempi di raddoppio.

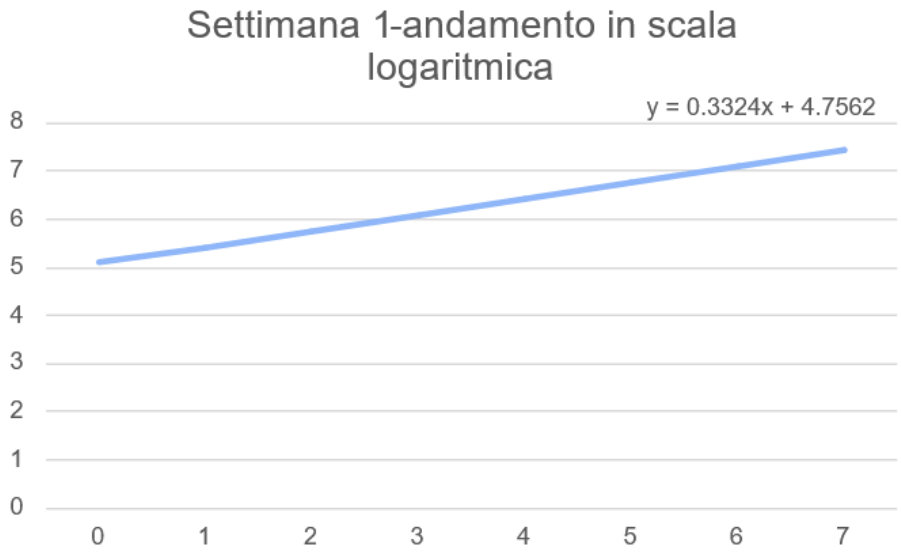


Figura 7.11: Modellizzazione lineare dei dati.

e $q = \ln(a)$, è stato possibile trovare anche i valori di m, q e t_2, quest ultimo ricalcolato analogamente al punto precedente ($t_2 = \frac{\ln 2}{m}$), riportati nella Tabella 7.6. La linearizzazione è utile quando si maneggiano dati molto grandi, in quanto "comprime" il grafico.

Conclusioni

L'analisi condotta sui dati riguardanti l'andamento nazionale della pandemia da COVID-19 offre una visione più dettagliata e scientificamente supportata della situazione dei contagi.

Attraverso l'impiego di strumenti quali grafici lineari, funzioni esponenziali e logaritmiche, sono state esplorate le dinamiche e le variazioni del contagio nel corso del tempo: dallo studio del picco

massimo degli ospedalizzati, alle variazioni settimanali e all'analisi delle singole ondate pandemiche. Tutto ciò fornisce un quadro più obiettivo, evidenziando anche l'impatto delle misure adottate per contrastare la diffusione del virus.

L'utilizzo di modelli matematici (esponenziali e logaritmici) ha permesso di studiare il fenomeno in maniera oggettiva, permettendo di stimare grandezze importanti nei tentativi di contenimento di questa infezione (considerata su scala nazionale) come il tempo di raddoppio dei contagiati. Una simile analisi ha permesso quindi lo sviluppo di misure capaci di contrastare e "misurare" il grado di pericolosità dell' agente patogeno.

7.4 Esercitazione in Python

Scopo dell'esercitazione

Vogliamo utilizzare un linguaggio di programmazione di alto livello (`Python`), per studiare i dati di variazione di temperatura dal 1961 al 2022 disponibili sul sito della Food and Agriculture Organization of the United Nations e capire se la tematica del surriscaldamento climatico sia una fenomeno reale oppure no.

Si consiglia di redigere una relazione scientifica in cui vengano riportate le considerazioni fatte e i risultati ottenuti.

Dataset

Per scaricare i dati necessari allo svolgimento dell' esercitazione, collegarsi al sito della Food and Agriculture Organization of the United Nations (**FAOSTAT**):

1. https://www.fao.org;
2. cliccare su *Climate Change*;
3. cliccare su *Climate Indicators*;
4. aprire il link *Temperature Change*.

Proponiamo il seguente esercizio, guidato da una serie di considerazioni.

Utilizzando gli strumenti disponibili sul sito tramite le schede `'DOWNLOAD DATA'` e `'VISUALIZE DATA'`, viene presentata una prima esplorazione dei dati che costituiranno la base della nostra analisi. Lo studio di quanto presentato in queste schede costituirà l'analisi preliminare della problematica affrontata.

Il dataset comprende una serie di informazioni climatiche, quali temperatura, precipitazioni e umidità, provenienti da diversi paesi e registrate nel corso del tempo. I dati sono rappresentati in varie unità di misura a seconda della variabile considerata; ad esempio, la temperatura viene espressa in gradi Celsius o Fahrenheit, mentre le precipitazioni potrebbero essere misurate in millimetri o pollici. Il periodo temporale coperto dal dataset può essere individuato analizzando le date di inizio e fine disponibili. I dati sono presentati con una certa periodicità, che può essere giornaliera, mensile, annuale o di altro tipo, in base alla fonte e alla natura dei dati raccolti. Infine, il dataset contiene informazioni provenienti da un numero variabile di paesi, che può essere identificato tramite l'analisi dell'elenco dei paesi rappresentati nei dati.

Download del dataset

Per scaricare i dati di nostro interesse, è necessario seguire le seguenti istruzioni:

1. cliccare sul link *All Data* come mostrato in Figura 7.12 e scegliere dove salvare il file *.zip* nel computer;

2. nella cartella in cui abbiamo scaricato il file .zip, estrarre tutti i dati contenuti nel *file archivio* (click destro sulla cartella → estrai tutto);

3. l'unico file che utilizzeremo è il file:
   Environment_Temperature_change_E_All_Data_NOFLAG.csv

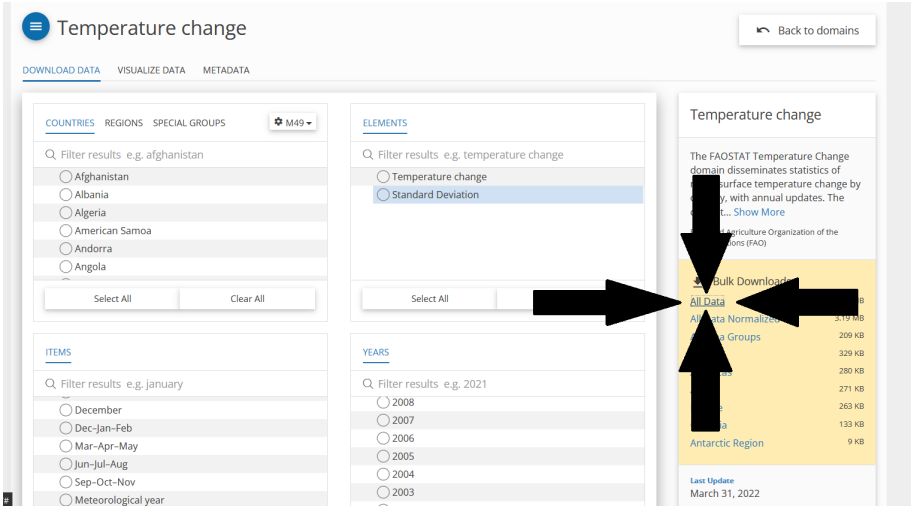


Figura 7.12: Link di download dei dati

Elementi di statistica

Per rendere più comprensibili i concetti che sono presentati in questa seconda esercitazione, richiamiamo qui alcuni elementi basilari di statistica.

Non verranno fatte dimostrazioni matematiche rigorose, ma verranno richiamati i concetti di:

- popolazione statistica;
- distribuzione statistica;
- mediana;
- media;
- deviazione standard.

Per una trattazione più approfondita, consultare il libro di testo [1].

Popolazione statistica In statistica, una popolazione è un insieme da cui vengono estratti i dati per l'analisi. Può consistere in un gruppo di individui, un insieme di oggetti, eventi, ecc...

Distribuzione statistica Una distribuzione statistica, o distribuzione di probabilità, descrive come sono distribuiti i valori di una certa proprietà (detta *carattere* o *modalità*) di una popolazione statistica (es. altezza degli individui).

Mediana È il valore che occupa il posto centrale in una serie di dati disposti in ordine crescente o decrescente. Se la serie è composta da un numero pari di elementi, la mediana è data dalla media aritmetica dei due dati centrali.

Media La media aritmetica di un insieme di dati numerici è definita come quel valore che si ottiene addizionando i valori tra loro e dividendo la somma ottenuta per il numero di dati raccolti.

Deviazione standard In statistica, la deviazione standard rappresenta un indice di quanto i valori di una distribuzione si discostino dalla media (per la definizione precisa si veda [1]).

Codice

Di seguito riportiamo il codice che ci consentirà di elaborare, visualizzare e manipolare i dati contenuti nel file scaricato, al fine di estrarre informazioni rilevanti da una popolazione statistica.

Una *libreria* è semplicemente un insieme di parti di codice già scritto contenente funzioni e strutture dati predefinite, predisposte per essere collegate ad altro codice attraverso opportuni collegamenti.

Nel nostro caso ci focalizziamo su 4 librerie come mostrato nel Codice 7.1:

- *NumPy*: numpy (*Numerical Python*) è la libreria fondamentale per il calcolo scientifico e matematico in Python.
- *Pandas*: pandas (*Panel Data / Python Data Analysis*) è una libreria creata per lavorare con insiemi di dati e presenta funzioni utili per analizzare, ripulire, esplorare e manipolare i dati.

- *Matplotlib.pyplot*: matplotlib (*Mathematical Plot Library*) è una libreria per creare visualizzazioni e grafici in Python.
- *io*: io (*input/output*) fornisce la possibilità di gestire l'input e l'output del nostro codice.

```
# Librerie principali per manipolare i dati
import numpy as np
import pandas as pd

# Librerie per la visualizzazione dei dati
import matplotlib.pyplot as plt

# Librerie per la gestione dell' input-output
import io
```

Codice 7.1: Librerie da importare.

Importazione di file in Google Colab

`Google Colab` è una piattaforma di cloud computing gratuita offerta da Google che consente agli utenti di eseguire codice Python in un ambiente basato sul cloud, senza dover configurare o gestire l'infrastruttura di calcolo. Questo strumento è ampiamente utilizzato per lo sviluppo e l'esecuzione di codice Python, l'analisi dati, il machine learning e altre attività di calcolo intensivo. Le ragioni del suo utilizzo per i nostri scopi includono l'accessibilità, la facilità d'uso e la possibilità di collaborare facilmente su progetti condivisi.

Una volta definite le librerie utili, dobbiamo caricare il file che abbiamo precedentemente scaricato su Google Colab. Per fare questo, utilizziamo l'oggetto *files* presente nella libreria *google.colab*, come mostrato nel Codice 7.2.

```
### Upload del file.
    from google.colab import files
    temp_file = files.upload()
```

Codice 7.2: Upload file.

Selezionando il file corretto, questo verrà caricato sul progetto su cui stiamo lavorando.

Lettura del dataset e struttura del dataset

Una volta che il file è stato correttamente caricato, dobbiamo leggerne il contenuto. Per fare questa operazione, ci avvaliamo di due librerie che abbiamo precedentemente importato:

- utilizziamo la funzione *BytesIO* della libreria *io* per indicare la posizione di memoria in cui è stato allocato il file dopo l'upload.
- utilizziamo la funzione *read_csv* di *pandas* per leggere il dataset.

Il risultato di queste due operazioni viene memorizzato su un elemento che chiamiamo "*temp_df*" (Codice 7.3).

```
1    ### Allocazione di memoria del file.
2        csv_name =
             'Environment_Temperature_change_E_All_Data_NOFLAG.csv'
3        my_file = temp_file[csv_name]
4        temp_df = pd.read_csv(io.BytesIO(my_file),
             encoding='latin-1')
```

Codice 7.3: Lettura del dataset.

Innanzitutto, è necessario esaminare la struttura del dataset che abbiamo importato.

Per svolgere questa analisi, utilizziamo la funzione *shape*, la quale ci restituisce l'output rappresentante la "forma" del *dataframe* creato. Inoltre, tramite l'oggetto *columns*, possiamo visualizzare i nomi delle colonne del dataframe. Infine, è possibile anche visualizzare l'intero dataframe caricato.

Queste operazioni vengono rappresentate nel Codice 7.4.

```
        # Struttura del dataframe importato
        print(f"Struttura del dataframe: {temp_df.shape[0]}
            righe e {temp_df.shape[1]} colonne.")

        # Nome delle colonne del dataframe
        print("\n Nome delle colonne nel dataframe:\n",
            temp_df.columns)

        # Visualizzazione dell'intero set di dati
        temp_df
```

Codice 7.4: Struttura del dataframe.

Esaminiamo la struttura del dataframe, che include il numero di colonne e righe, al fine di verificare la coerenza con le informazioni ottenute durante l'analisi preliminare. Questo ci permette di valutare se i dati caricati riflettano fedelmente le aspettative e le caratteristiche della nostra indagine.

Manipolazione dei dati

Come già visto, un'altra caratteristica della libreria *pandas* è quella di manipolare i dati. Ad esempio è possibile:

- selezionare specifici elementi del dataframe tramite la funzione "*loc*";
- eliminare informazioni superflue presenti nei dati utilizzando "*drop*";
- rinominare le colonne di interesse con "*rename*";
- controllare valori mancanti all'interno del dataframe con "*isnull*".

Applichiamo quanto visto al nostro insieme di dati e verifichiamo quanti elementi nulli sono presenti (Codice 7.5).

```
# Seleziono gli elementi del dataframe che non contengono i valori della deviazione standard
temp_df = temp_df.loc[temp_df.Element == 'Temperature change']

# Elimino le colonne non necessarie
temp_df.drop(columns=['Area Code', 'Area Code (M49)', 'Months Code', 'Element Code', 'Element', 'Unit'], inplace=True)

# Modifico il nome delle colonne, rimuovendo la Y davanti all'anno
temp_df.rename(columns={x:x[1:] for x in temp_df.columns if 'Y' in x}, inplace=True)

# Modifico il nome della colonna "Area" in "country_name", e la colonna "Months" in "months"
temp_df.rename(columns={'Area': 'country_name', 'Months':'months'}, inplace=True)

# Verifico se nel dataframe esistono dei valori nulli
count_nan = temp_df.isnull().sum().sum()
print('Numero di elementi nulli: ', str(count_nan))
```

Codice 7.5: Alcune funzioni per manipolare il dataframe.

Tipologia di dati, valori univoci e percentuale di dati mancanti

Costruiamo un secondo dataframe a partire dai dati che sono contenuti nel primo. In particolare, ci prefiggiamo di avere un dataframe avente come indice di riga gli anni in cui sono state fatte le registrazioni e come colonne il tipo di oggetto che vi è contenuto, il numero di valori univoci che si trovano nel dataset e la percentuale di dati mancanti.

A questo proposito costruiamo il nuovo dataframe partendo da un *dizionario* come descritto nel Codice 7.6.

```
### Creare un DataFrame usando un dizionario.

diz = {"Dtype": temp_df.dtypes,
       "Unique values": temp_df.nunique(),
       "Missing values(%)":
           (temp_df.isnull().sum()/temp_df.shape[0])*100
       }
df_temp_info= pd.DataFrame(diz).rename_axis('Columns',
    axis='rows')
```

Codice 7.6: Costruzione di un dataframe da un dizionario.

Per visualizzare graficamente in che modo sono distribuite le percentuali di questi dati mancanti dal 1961 al 2022 (Codice 7.7), utilizziamo la libreria grafica *"matplotlib.pyplot"* importata precedentemente.

```
### Creare un grafico da un insieme di dati.

df_temp_info2 = df_temp_info.copy().iloc[2: , :]
df_temp_info2.index.rename('Year', inplace=True)
ts = [t.timestamp() for t in
    pd.to_datetime(df_temp_info2.index)]
tus = 1
lus = 2
ts_ticks = ts[::tus]
ts_lbl = [ t.strftime('%Y') for t in
    pd.to_datetime(df_temp_info2.index) ]
ts_lbl = ts_lbl[::tus]
ts_lbl = [ t if i%lus==0 else '' for i, t in
    enumerate(ts_lbl)]
fig, ax = plt.subplots(1, 1, sharex=True, figsize=(8,
    8), dpi = 100)
ax.scatter(ts, df_temp_info2['Missing values(%)'],
    label=f"Percentuale di valori mancanti")
ax.legend()
```

```
    ax.set_xticks(ts_ticks)
    ax.set_xticklabels(ts_lbl, rotation=45, ha='right')
    ax.xaxis.grid()
    ax.set_xlabel('Anno')
    ax.set_ylabel('Percentuale')
    plt.title(f"Percentuale dei valori nulli nei dati",
        fontsize=12)
    plt.show()
```

Codice 7.7: Utilizzo della grafica.

Basandoci sul grafico ottenuto, possiamo vedere come la popolazione statistica di valori mancanti possa essere suddivisa in due gruppi: prima del 1992 e dopo il 1992. Calcoliamo quali sono la media, la mediana e la deviazione standard della percentuale di dati mancanti in questi due gruppi. Nel Codice 7.8 viene eseguito il calcolo solo per il primo gruppo (fino al 1992).

```
    ### Calcolo di variabili statistiche.
    # Popolazione dei dati prima del 1992 (escluso)
    df_temp_first_half = df_temp_info2[df_temp_info2.index
        < '1992']
    # Calcoliamo la media, mediana e deviazione standard
    df_temp_first_half_media = df_temp_first_half['Missing
        values(%)'].mean()
    df_temp_first_half_mediana =
        df_temp_first_half['Missing values(%)'].median()
    df_temp_first_half_std_dev =
        df_temp_first_half['Missing values(%)'].std()
```

Codice 7.8: Calcolo della media, mediana e deviazione standard per il primo raggruppamento.

Le stesse quantità statistiche (media, mediana e deviazione standard) possono essere calcolare per il secondo gruppo di dati, ovvero dal 1992 fino al 2022.

Ripulire il dataframe

Allo scopo di ottimizzare la struttura del dataframe, intendiamo procedere con una serie di operazioni di riorganizzazione. Tali operazioni sono dettagliate nel Codice 7.9 e descritte in seguito:

- individuiamo gli indici delle righe che contengono almeno un elmento nullo dal 1961;
- eliminiamo le righe che hanno come indici i valori trovati;
- sostituiamo le stagioni corrispondenti ai rispettivi mesi.

```
### Pulizia del DataFrame.
# Individuiamo gli indici delle righe che contengono almeno un elemento nullo dal 1961
index_nan = temp_df.loc[temp_df.isnull().any(axis=1), '1961':].index
# Eliminiamo le righe che hanno come indici i valori trovati
temp_df.drop(index_nan, inplace=True)
# Sostituiamo le stagioni corrispondenti ai rispettivi mesi
temp_df.months.replace({
    'Mar\x96Apr\x96May': 'Spring',
    'Jun\x96Jul\x96Aug': 'Summer',
    'Sep\x96Oct\x96Nov': 'Fall',
    'Dec\x96Jan\x96Feb': 'Winter',
    }, inplace=True)
```

Codice 7.9: Riorganizzazione del dataframe.

Andamento delle temperature nei singoli stati per l'anno metereologico

Grazie alle operazioni eseguite nei paragrafi precedenti, abbiamo ottenuto una struttura del dataframe che si dimostra idonea per lo scopo dell'esercitazione. Tale obiettivo consiste nell'analizzare dati

sperimentalmente registrati al fine di valutare l'effettiva presenza di cambiamenti climatici.

Consideriamo quindi almeno tre stati che sono presenti all'interno del dataframe che abbiamo chiamato "*temp_df*". Possiamo visualizzare questi stati utilizzando le righe di Codice 7.10.

```
### Ottenere una lista di valori univoci da una colonna.
lista_stati = temp_df['country_name'].unique()
```

Codice 7.10: Lista di valori univoci estratti da una colonna.

Ci interessiamo ai dati registrati nelle ultime quattro decadi, ovvero dal 1982, e, per ciascuno stato, facciamo un grafico dell'andamento della variazione della temperatura registrata nell'anno metereologico [8] come descritto nel Codice 7.11.

```
### Grafico per N=40 anni
N = 40
df_last_N_column_italy = pd.DataFrame()
for country, df_country in temp_df.groupby('country_name'):
    if country == "Italy":
        titolo = f'Andamento della variazione delle temperature in {country}'
        xlabel = 'Anno'
        ylabel = 'Variazione anno metereologico'
        kind = 'line'
        df_country.index = df_country.months
        df_last_N_column_italy = df_country.iloc[: , -N:]
        df_last_N_column_italy.loc['Meteorological year'].plot(
            title = titolo,
            xlabel = xlabel,
            ylabel = ylabel,
            kind = kind,
```

[8] Nell'esempio il paese scelto è l'Italia, ma è possibile cambiare il codice in funzione del paese scelto.

```
17                grid = True,
18                rot = 45,
19                figsize = (12,8))
20    plt.show()
```

Codice 7.11: Andamento annuale della variazione delle temperature.

Osserviamo che :

- variando il parametro N, variamo l'anno da cui parte il grafico: rappresenta infatti le ultime N colonne del dataframe;
- utilizziamo l'oggetto *plot* della libreria *Pandas* per fare il grafico di nostro interesse e non il pacchetto della libreria *Matplotlib*. Esistono quindi diverse modalità per poter raggiungere lo stesso risultato.

Analizziamo l'iterazione del ciclo "for" presente nel Codice 7.11 e in particolare ci soffermiamo sull'utilità della clausola "if" al suo interno.

Esaminiamo anche i grafici ottenuti, fornendo una descrizione dettagliata di ciascuno e specificando a quale paese si riferiscono. Durante l'analisi dei grafici, osserviamo l'andamento generale, verificando se sia crescente o decrescente. Successivamente, individuiamo la data intorno alla quale si registra la variazione maggiore per ciascun paese. Confrontando queste variazioni, identifichiamo il paese che ha mostrato la maggiore variazione nel periodo considerato.

Andamento stagionale delle temperature nei singoli stati

Ci proponiamo in questa sezione di studiare i dati in modo più dettagliato, considerando l'andamento stagionale delle variazioni delle temperature.

Scegliamo i tre stati considerati nella sezione precedente.

Consideriamo sempre i dati registrati dal 1982 e rappresentiamo su un grafico l'andamento invernale, primaverile, estivo e autunnale (per ciascun paese) come descritto nella prima parte del Codice 7.12 (fino a riga 16).

```
1     ### Grafico dell'andamento stagionale delle temperature
2     M = 40
3     season_list = ['Winter', 'Spring', 'Summer', 'Fall']
4     for country, df_country in
          temp_df.groupby('country_name'):
5       if country == 'Italy':
6         df_country.index = df_country.months
7         df_last_IT = df_country.iloc[: , -M:]
8         df_season_IT =
              df_last_IT.loc[df_last_IT.index.isin(season_list)]
9         df_season_IT.T.plot(title = f'Andamenti stagionali
              per {country}',
10                            xlabel = 'Anno',
11                            ylabel = 'Variazione Stagionale',
12                            kind = 'line',
13                            grid = True,
14                            rot = 45,
15                            figsize = (8,8))
16
17        df_mean =
              df_season_IT.groupby(np.arange(len(df_season_IT.columns))//4,
              axis=1).mean()
18        df_mean.plot(title = f'Andamento quadriennale in
              {country} per stagione',
19                   xlabel = '10 anni a blocco',
20                   ylabel = 'Variazione media temperature
                         quadriennale',
21                   kind = 'bar',
22                   grid = True,
23                   rot = 45,
24                   figsize = (8,8)).legend(loc='center
                         left',bbox_to_anchor=(1.0, 0.5))
```

Codice 7.12: Andamento stagionale della variazione delle temperature.

Esaminiamo l'andamento dei dati ottenuti e valutiamo se sia consistente con quanto evidenziato nel paragrafo precedente. Successivamente, individuiamo gli anni in cui si sono verificate le variazioni di temperatura più significative per ogni stagione e per ciascun paese.

Andamento delle temperature nel mondo

In questa sezione, esploriamo se i cambiamenti climatici siano un fenomeno locale, limitato agli stati singoli considerati, oppure se debba essere interpretato come un fenomeno globale.

Per valutare questa prospettiva, esaminiamo i dati globali dal 1961 ad oggi e ripetiamo l'analisi condotta precedentemente. Nel Codice 7.13, illustriamo il processo di selezione delle sole righe pertinenti ai dati globali.

```
1 ### Filtraggio dei dati registrati per il pianeta
      Terra.
2 world_temp_df = temp_df.loc[temp_df.country_name ==
      'World']
```

Codice 7.13: Filtraggio dei dati del pianeta.

Concludiamo questa sezione analizzando i grafici ottenuti e confrontando i risultati con le informazioni precedentemente discusse. Questo ci consentirà di trarre conclusioni più complete sull'entità e la portata dei cambiamenti climatici considerati a livello locale e globale.

Conclusioni

Alla conclusione dello studio, è essenziale includere un paragrafo che riassuma e amplifichi le considerazioni emerse. In primo luogo,

sarà importante delineare la problematica affrontata, fornendo contesto e motivazioni per lo sviluppo del codice. Successivamente, si potrà esaminare in dettaglio il metodo adottato per affrontare tale problema, includendo una descrizione del tipo di studio condotto e delle strategie implementate. Attraverso un'analisi approfondita dei dati e dei grafici ottenuti, sarà possibile elaborare conclusioni significative che aiutino a chiarire il quadro complessivo e a trarre implicazioni pertinenti.

Questo approccio consentirà di fornire una sintesi esaustiva e riflessiva del lavoro svolto, evidenziando l'importanza e le implicazioni delle analisi condotte.