

Indice

Prefazione	XIII
A chi si rivolge questo testo	XIII
Il contenuto e la sua organizzazione	XV
Introduzione	1
1 Metodo di studio, metodo di lavoro	2
2 Open data	2
3 Cosa non si impara	4
1 Strumenti e primi passi	5
1.1 R e RStudio	5
1.1.1 Linguaggio R	5
1.1.2 RStudio	6
1.1.3 RStudio Cloud	7
1.1.4 Package manager	7
1.1.5 Package Tidyverse	8
1.2 Python e strumenti	9
1.2.1 Scelta A: Anaconda Distribution	10
1.2.2 Scelta B: Installazione manuale	10
1.2.3 Google Colab	12
1.2.4 Package NumPy e pandas	12
1.3 Editor plain text evoluti	12
1.4 Formato CSV	13
2 Statistiche descrittive	15
2.1 Analisi dei valori mancanti	15
2.2 R: statistiche descrittive e funzioni di utilità	16
2.3 Python: statistiche descrittive e funzioni di utilità	19
3 Organizzazione dei dati e operazioni su data frame	21
3.1 Lettura di un dataset CSV	23
3.1.1 Errori di lettura	24
■ Errore di nome o percorso del dataset	24
■ Errore di separatore	25
3.2 R: Selezione di colonne	27
■ Funzione <code>select()</code>	28

3.2.1	Altre modalità di selezione	29
	▪ Selezione per posizione delle colonne	29
	▪ Selezione per intervallo	30
	▪ Selezione per negazione	30
	▪ Selezione con <i>selection helper</i>	33
3.3	R: Rinominare e spostare colonne	34
	▪ Funzioni <code>rename()</code> e <code>relocate()</code>	35
3.4	R: Slicing, creazione e cancellazione di colonne	37
3.4.1	Slicing	37
3.4.2	Creazione	39
3.4.3	Cancellazione	40
3.4.4	Colonne con valori calcolati	41
	▪ Funzione <code>mutate()</code>	42
3.5	R: Separazione e unione di colonne	43
3.5.1	Separazione	43
	▪ Il codice fiscale	44
	▪ Date	46
3.5.2	Unione	47
3.6	R: Ordinamento di data frame	48
3.6.1	Ordinamento di più colonne	48
3.6.2	Ordinamento con valori da una lista	49
	▪ Caso dei nomi di mesi	49
3.7	R: Pipe	51
3.8	Python: Selezione di colonne	54
3.8.1	Selezione di colonne nella lettura del dataset	56
3.8.2	Selezione di colonne da un data frame	57
3.8.3	Selezione per posizione, per intervallo e con <i>selection helper</i>	58
3.8.4	Selezione per negazione	58
3.9	Python: Rinominare e spostare colonne	61
	▪ Funzioni <code>rename()</code> e <code>reindex()</code>	62
3.10	Python: Slicing con indice, creazione e cancellazione di colonne	63
3.10.1	Slicing di vettori/matrici NumPy	63
3.10.2	Slicing di data frame pandas	64
	▪ Metodi <code>.loc</code> e <code>.iloc</code>	66
3.10.3	Selezione con <i>selection helper</i>	69
3.10.4	Creazione con modo standard	70
3.10.5	Cancellazione	71
	▪ Funzioni <code>insert()</code> e <code>assign()</code>	71
3.11	Python: Separazione e unione di colonne	72
3.11.1	Separazione	72
3.11.2	Unione	75
3.12	Python: Ordinamento di data frame	75
3.12.1	Ordinamento di colonne	75
3.12.2	Ordinamento con indice	76

3.12.3	Trasformazione di un data frame da indicizzato a non indicizzato	78
3.12.4	Trasformazione di una colonna in indice	78
3.12.5	Ordinamento con valori da una lista	78
	■ Caso del nome dei mesi	79
3.13	Python: Composizione di operazioni	81
4	Condizioni logiche e selezione di righe	83
4.1	Operatori logici	85
4.2	R: Selezione di righe con condizione logica	86
	■ Funzione <code>filter()</code>	86
4.2.1	Maschera booleana	87
4.2.2	Esempi di selezione con condizioni logiche	90
4.3	Python: Selezione di righe con condizione logica	99
4.3.1	Selezione di righe: modo standard	100
4.3.2	Selezione di righe con <code>query()</code>	104
5	Operazioni su date, stringhe e valori mancanti	105
5.1	R: Operazioni su date e su stringhe	107
5.1.1	Data e tempo	107
5.1.2	Selezione con condizione logica su date	110
5.1.3	Stringhe	111
5.2	R: Gestione dei valori mancanti e trasformazioni di tipo	113
	■ Funzione <code>is.na()</code>	113
	■ Funzioni <code>any()</code> e <code>colSums()</code>	114
5.2.1	Verificare la presenza e contare i valori mancanti	113
5.2.2	Sostituzione di valori mancanti: modo standard	115
	■ Funzione <code>replace_na()</code>	116
5.2.3	Eliminare le righe contenenti i valori mancanti	117
5.2.4	Trasformazioni di tipo di dato	119
5.3	R: Esempio con date, stringhe e valori mancanti	119
5.3.1	Eurostat: Final energy consumption	119
5.3.2	Open Data Berlin: Fahrraddiebstahl in Berlin	123
5.3.3	Quando una «mano invisibile» modifica i nostri dati	129
5.3.4	Trasformazione con funzioni di base	130
5.3.5	Soluzione corretta	132
5.3.6	Trasformazione con funzioni specializzate	133
	■ Funzione <code>parse_date_time()</code>	133
5.3.7	Verifica dei risultati dei due metodi	135
5.4	Python: Operazioni su date e su stringhe	136
5.4.1	Data e tempo	136
	■ Funzione <code>pandas.to_datetime()</code>	136
	■ Funzioni <code>datetime.datetime.strptime()</code>	138
	■ Funzione <code>datetime.datetime.strftime()</code>	139
5.4.2	Configurazione internazionale	139

5.4.3	Operazioni su colonne di data frame in formato data/orario	141
5.4.4	Selezioni con condizione logica su date	143
5.4.5	Stringhe	144
5.5	Python: Gestione dei valori mancanti e trasformazioni di tipo	144
5.5.1	Funzioni per gestire i valori mancanti	144
5.5.2	Modifica su una vista o su una copia	146
	■ Funzioni <code>np.replace()</code> e <code>pd.fillna()</code>	149
5.5.3	Selezionare le righe che non contengono dei valori mancanti	149
5.5.4	Trasformazioni di tipo di dato	150
5.6	Python: Esempi con date, stringhe e valori mancanti	151
5.6.1	Eurostat: Final energy consumption	151
5.6.2	Open Data Berlin: Fahrraddiebstahl in Berlin	154
6	Operazioni di pivoting e trasformazioni <i>wide-long</i>	159
6.1	R: Operazioni di pivoting	162
6.1.1	Da <i>long</i> a <i>wide</i>	162
6.1.2	Nazioni Unite: World Population Prospects 2022	163
6.1.3	Da <i>wide</i> a <i>long</i>	165
6.1.4	GOV.UK: Gender pay gap	166
6.2	Python: Operazioni di pivoting	169
6.2.1	NYC Open Data: Youth Behavior Risk Survey	169
6.2.2	Da <i>wide</i> a <i>long</i> con colonne	170
6.2.3	Da <i>long</i> a <i>wide</i> con colonne	172
6.2.4	<i>Wide-long</i> con livelli di indice	174
6.2.5	ISTAT: Spostamenti per studio o lavoro	174
	■ Problemi di codifica di caratteri: UTF-8 vs. ISO-8859-1	174
	■ Problema di separatore anomalo	176
	■ Data frame indicizzato	177
6.2.6	Trasformazioni	178
	■ Funzione <code>unstack()</code>	178
	■ Funzione <code>stack()</code>	178
6.2.7	Da <i>long</i> a <i>wide</i> con valori numerici	179
7	Gruppi e operazioni su gruppi	187
7.1	R: Gruppi, aggregazioni e trasformazioni	188
7.1.1	ISTAT: Previsioni della popolazione residente per sesso, età e comune	189
	■ Funzione <code>group_by()</code>	191
	■ Dettagli di un indice	193
7.1.2	Indicizzazioni per gruppo e aggregazioni	195
	■ Funzioni <code>group_by()</code> e <code>summarize()</code>	195
	■ Contare le righe: funzione <code>n()</code>	196
	■ Media aritmetica: funzione <code>mean()</code>	197
	■ Valori massimo e minimo: funzioni <code>max()</code> e <code>min()</code>	198
	■ Elenco di funzioni di aggregazione	198

7.1.3	Indicizzazione di gruppi con più variabili	199
7.1.4	Operazioni di aggregazione con più livelli di indice	200
7.1.5	Ordinamenti di risultati aggregati	202
7.1.6	Creazione di colonne e data frame indicizzati	204
7.1.7	Slicing su risultati aggregati	206
	▪ Funzioni <code>slice_*</code> ()	207
	▪ Combinazione <code>filter()</code> e <code>rank()</code>	209
7.1.8	Colonne calcolate con valori di gruppo	214
7.2	Python: Gruppi, aggregazioni e trasformazioni	216
7.2.1	Bureau of Transportation Statistics: voli di gennaio 2022	216
7.2.2	Indicizzazione di gruppi e aggregazioni	218
	▪ Funzioni <code>groupby()</code> e <code>aggregate()</code>	218
	▪ Contare le righe per gruppo, media aritmetica e somma	218
7.2.3	Multi-indice di riga e colonna	220
7.2.4	Nomi delle colonne con valori aggregati	221
7.2.5	Ordinamento di colonne	222
	▪ Altre funzioni di aggregazione	223
7.2.6	Ordinamenti su più colonne	224
7.2.7	Ordinamenti sui livelli di un indice	224
7.2.8	Selezione su risultati aggregati	225
	▪ Funzioni <code>nlargest()</code> e <code>nsmallest()</code>	227
7.2.9	Colonne calcolate con valori di gruppo	228
7.2.10	Ordinamenti interni ai gruppi	230
8	Istruzioni condizionali e iterazioni	233
8.1	R: Istruzioni condizionali e iterazioni	234
8.1.1	Condizioni	234
	▪ Funzione <code>ifelse()</code>	234
	▪ Funzione <code>if_else()</code>	237
	▪ Funzione <code>case_when()</code>	238
	▪ Funzione <code>if()</code> e costrutti <code>if-else</code> e <code>if-else if-else</code>	239
8.1.2	Iterazioni	239
8.1.3	Iterazioni innestate	241
8.2	Python: Istruzioni condizionali e iterazioni	245
8.2.1	Istruzioni condizionali	245
	▪ Funzione <code>if()</code>	245
	▪ Costrutti <code>if-else if-elif-else</code>	246
	▪ Funzione <code>np.where()</code>	246
	▪ Funzione <code>np.select()</code>	247
	▪ Funzioni <code>pd.where()</code> e <code>pd.mask()</code>	250
8.2.2	Iterazioni	252
8.2.3	Iterazioni innestate	254
8.2.4	Iterazioni su multi-indici	257
	▪ Uso di <code>for()</code> e <code>join()</code>	260
	▪ Uso di <code>for loop</code> e <code>iteritem()</code>	261

9	Funzioni e operazioni multicolonna	263
9.1	R: Definizione di funzioni e funzioni anonime	264
9.1.1	Usò di funzioni	265
9.1.2	Data masking	266
9.1.3	Funzioni anonime	269
9.2	R: Operazioni multicolonna	270
9.2.1	Metodo base	270
	▪ Funzioni <code>apply()</code> , <code>lapply()</code> , <code>sapply()</code>	270
9.2.2	Mapping	274
	▪ Funzione <code>map()</code> e varianti	274
9.2.3	Mapping e funzioni anonime	275
9.2.4	Mapping condizionale	277
9.2.5	Selezione di righe multicolonna	279
9.2.6	Trasformazioni multicolonna	280
9.2.7	Usò dei valori mancanti	281
9.2.8	Esempi e tempi d'esecuzione	281
9.3	Python: Definizione di funzioni e funzioni lambda	285
9.3.1	Funzioni	285
9.3.2	Funzioni lambda	287
9.4	Python: Operazioni multicolonna	288
9.4.1	Casi d'usò	290
9.5	Python: Funzionalità avanzate su gruppi e su ordinamenti	294
9.5.1	Funzionalità avanzate per gruppi	294
	▪ Funzioni <code>groupby()</code> e <code>apply()</code>	294
9.5.2	Ordinamenti speciali con <i>lambda function</i> e <i>naturally sorting</i>	296
	▪ Ordinamento con <i>key</i> e <i>lambda function</i>	298
	▪ <i>Naturally sorting</i>	299
10	Join di data frame	301
10.1	Concetti di base	302
10.1.1	Chiavi di una operazione di join	303
10.1.2	Tipi fondamentali di join	303
10.2	R: Join di data frame	304
10.2.1	Funzioni di join	307
	▪ Funzione <code>inner_join()</code>	307
	▪ Funzione <code>full_join()</code>	308
	▪ Funzioni <code>left_join()</code> e <code>right_join()</code>	309
	▪ Funzione <code>merge()</code>	310
10.2.2	Join e chiavi duplicate	310
	▪ Semi join: funzione <code>semi_join()</code>	316
	▪ Anti join: funzione <code>anti_join()</code>	318
10.3	Python: Join di data frame	319
	▪ Funzione <code>merge()</code>	321
	▪ Inner join	321

■ Outer join	323
■ Left join e right join	324
10.3.1 Caso con indici	324
10.3.2 Chiavi duplicate	326
■ Semi join	332
■ Anti join	333



11 Esempi completi di join, trasformazioni e risultati preliminari con open data

11.1 R: Esempio completo	
11.1.1 Ministero delle Infrastrutture e dei Trasporti: Immatricolazione autoveicoli nel mese di settembre 2022	
11.1.2 ISTAT: Dataset Codici Entità Territoriali	
11.1.3 Operazioni preliminari	
11.1.4 Join tra immatricolazioni e codici delle unità territoriali	
11.1.5 Operazioni di trasformazione	
11.1.6 Risultati	
11.1.7 Ministero dell'Economia e delle Finanze: Dataset Dichiarazioni dei redditi anno 2020	
11.1.8 Operazioni preliminari	
11.1.9 Soluzione 1: Join tra redditi ed entità territoriali con codice catastale	
11.1.10 Soluzione 2: Join tra immatricolazioni ed entità territoriali con redditi con sigla automobilistica	
11.1.11 ISTAT: Dataset Popolazione province al 1° gennaio 2022	
11.1.12 Selezione delle Province	
11.1.13 Join tra immatricolazioni, redditi e popolazione	
11.1.14 Risultati	
11.2 Python: Esempio di join di data frame	
11.2.1 Nazioni Unite: popolazione mondiale	
11.2.2 U.S. Energy Information Administration (EIA): Total energy consumption	
11.2.3 Our World in Data/Ember: Electricity generation	
11.2.4 Join tra consumo di energia e popolazione mondiale	
11.2.5 Join tra consumo di energia+popolazione mondiale e produzione di energia elettrica	
11.2.6 Risultati	
11.2.7 U.S. Energy Information Administration: Emissioni 2020	
11.2.8 World Resource Institute: Global Power Plant Database	
11.2.9 Risultati	

12 Dati in formato lista/dizionario

12.1 R: Dati in formato lista	
12.1.1 Liste come colonne di un data frame	

- 12.1.2 Trasformazione di una colonna definita come lista in righe o colonne di data frame
- 12.2 R: Casi d'uso con dati in formato JSON
 - 12.2.1 NOAA - Climate at a Glance: Global Time Series
 - 12.2.2 Eventi storici italiani
 - 12.2.3 The Nobel Prize
 - 12.2.4 Politici italiani
 - 12.2.5 Dataset JSON di grandi dimensioni
- 12.3 Python: Dati in formato dizionario
 - 12.3.1 Metodi per formato dict
 - 12.3.2 Da formato dict a formato data frame con un livello di innestamento
 - 12.3.3 Da formato dict a formato data frame con più livelli di innestamento
 - 12.3.4 Da formato dict a formato data frame con più livelli di gerarchia ed elenchi
- 12.4 Python: Casi d'uso con dati in formato JSON
 - 12.4.1 Politici italiani
 - 12.4.2 The Nobel Prize