

# VisiData: il coltellino svizzero per i dati, che probabilmente non conosci

VisiData è un fantastico strumento open source per esplorare e manipolare dati. Mette insieme la chiarezza di un foglio di calcolo, con l'efficienza del terminale e la potenza di Python, in una utility leggera in grado di gestire milioni di righe con facilità.

**Autore: Andrea Borruso**

Sito web: <https://gbvitrano.github.io/guidaVisiData/>

Repo: <https://github.com/gbvitrano/guidaVisiData/>

By @ondatit | Licenza CC-BY-SA-4.0



# VisiData: il coltellino svizzero per i dati, che probabilmente non conosci

## I Guida VisiData

### 1 La guida

- 1.0.0.1 Un esempio di apertura di un file grande
- 1.0.0.2 Navigare tra i dati
- 1.0.1 I fogli

### 2 Lavorare sulle righe

- 2.0.0.1 Selezione di righe tramite espressione regolare
- 2.0.0.2 Selezione di righe tramite espressioni Python
- 2.0.0.3 Selezione di righe tramite espressioni Python e regex
- 2.0.0.4 Selezionare un numero random di righe
- 2.0.0.5 Spostare righe
- 2.0.0.6 Cancellare righe
- 2.0.0.7 Modificare il contenuto

### 3 Lavorare sulle colonne

- 3.0.0.1 Il foglio delle colonne
- 3.0.0.2 Impostare il tipo di campo
  - 3.0.0.2.1 Impostare numero di decimali in un campo floating
  - 3.0.0.2.2 Impostare formato data personalizzato
- 3.0.0.3 Rinominare le colonne
- 3.0.0.4 Rinominare le colonne in blocco
- 3.0.0.5 Come espandere, ridurre e rimuovere colonne
- 3.0.0.6 Spostare una colonna
- 3.0.0.7 Modificare una colonna tramite un'espressione Python
- 3.0.0.8 Calcoli per colonna
- 3.0.0.9 Come definire una colonna chiave
- 3.0.0.10 Manipolare le colonne dal foglio delle colonne
- 3.0.0.11 Creare una nuova colonna
  - 3.0.0.11.1 Nuova colonna tramite espressione
  - 3.0.0.11.2 Nuove colonne come risultato di splitting
  - 3.0.0.11.3 Nuove colonne come risultato di cattura e estrazione
- 3.0.0.12 Trasformare una colonna tramite espressione regolare (trova e sostituisci)
- 3.0.0.13 Il foglio delle frequenze

### 4 Lavorare sulle celle

- 4.0.0.1 Fare un trova e sostituisci globale

## 5 Ordinare e filtrare

- 5.0.0.1 Ordinare righe
- 5.0.0.2 Filtrare righe
  - 5.0.0.2.1 Filtro tramite espressione Python
  - 5.0.0.2.2 Filtro tramite espressione Python su campo datetime
  - 5.0.0.2.3 Filtri tramite espressioni python, basati su più colonne

## 6 Riepilogo dei dati

- 6.0.0.1 Aggiunta di aggregatori
- 6.0.0.2 Il riepilogo globale

## 7 Salvare un foglio

## 8 Fare il redirect dell'output verso lo stdout

## 9 Fare JOIN tra tabelle

## 10 Concatenare tabelle

## 11 Tabelle Pivot

## 12 Creare un grafico di dispersione (scatter plot)

## 13 Moduli Python

## 14 Salvare un flusso di lavoro

- 14.0.0.1 Introduzione
- 14.0.0.2 Applicare lo stesso flusso a un file diverso, ma omologo
- 14.0.0.3 Nota sul file di log

## 15 Soluzione problemi

- 15.0.0.1 Riga attiva di colore nero
- 15.0.0.2 Caratteri non leggibili in Windows Subsystem for Linux

## 16 Ricette

- 16.0.0.1 Salvare una tabella HTML in CSV, a partire da una pagina web

## 17 Fare proposte e/o Chiedere supporto

## 18 Note finali

## 19 URL consigliati

## II Esempi

## 20 10 Modi di usare Visidata

- 20.1 1. esplora i set di dati senza sforzo, indipendentemente dal formato
- 20.2 2. convertire da qualsiasi formato di input in qualsiasi formato di output
- 20.3 3. come sostituto interattivo di grep, awk, sed, cut, sort, uniq
- 20.4 4. grattare i dati della tabella HTML da una pagina Web:
- 20.5 5. vedere una differenza visiva di due fogli con la stessa struttura
- 20.6 6. esplora un oggetto dal Python REPL

- 20.7 7. esplora tutti i pandas DataFrame
- 20.8 8. creare una pipeline di dati ad hoc
- 20.9 9. rinominare un mucchio di file
- 20.10 10. vedi tutte le opzioni e i comandi disponibili

## 21 Come salvare un insieme di operazioni e applicarle come modello ad altri file

- 21.1 Introduzione
- 21.2 Applicare le stesse procedure a un file diverso, ma omologo
- 21.3 Nota sul file di log

## 22 Informativa privacy estesa

- 22.1 Cookie di terze parti
- 22.2 Widget Social Network
- 22.3 Pubblicità
- 22.4 Come disabilitare i cookie mediante configurazione del browser

# I. Guida VisiData

# 1 La guida

VisiData è un **foglio elettronico a riga di comando**. Potrebbe suonare come una contraddizione, perché si accosta qualcosa con interfaccia ricca (come un foglio elettronico) a qualcosa di visualmente molto minimale. In realtà questa è **una delle idee forti** di questa applicazione, che prende il meglio di questi due mondi: l'approccio a griglia e multi tabella del primo, e la rapidità, l'immediatezza, l'essere *easy* e "subito pronto" di un terminale.

Di base quindi bisogna prima aprire il terminale. Poi per aprire un file basta scrivere un comando con questa struttura:

```
vd nomeFile.csv
```

A seguire soltanto un piccolo estratto sul come usarlo. Per approfondire ho creato un elenco di [URL consigliati](#).

VisiData si usa al 99% con la tastiera, quindi è bene che il `Caps Lock` non sia attivo: gli *shortcut* da tastiera non funzionerebbero correttamente (grazie [Salvatore Fiandaca](#) (<https://twitter.com/totofiandaca>) per la nota).

## 1.0.0.1 Un esempio di apertura di un file grande

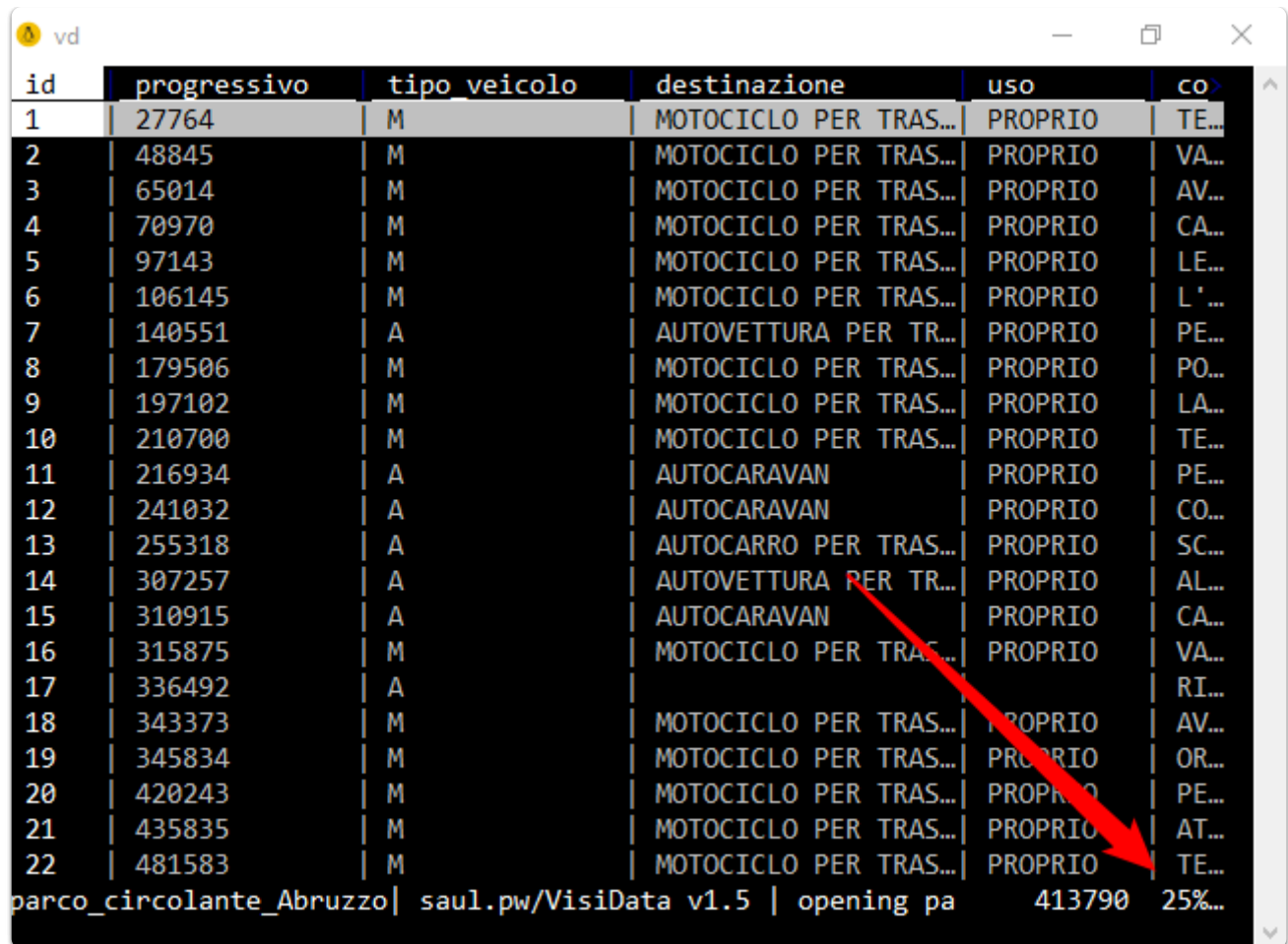
"Grande" in informatica non vuol dire niente, è come "salato" con un piatto di pasta.

Un CSV grande per il mio PC (un notebook del 2015, con processore i7 e 8 GB di RAM) - se voglio lavorarci in modo "visuale" con un foglio elettronico - potrebbe essere già un CSV da 500000 righe, che mi si apre dopo 1 minuto e 15 secondi e dove un "trova e sostituisci" diventa operazione molto lunga (uso LibreOffice Calc).

Ci sono anche limiti che non sono personali. Se per esempio provassi a lavorare con uno dei dataset del "[Parco Circolante dei veicoli](#) (<http://dati.mit.gov.it/catalog/dataset/parco-circolante-dei-veicoli>) " in Italia scoprirei dei limiti strumentali.

Un esempio con i dati dell'[Abruzzo](#) ([http://geodata.mit.gov.it/datasets/parco\\_circolante\\_Abruzzo.csv](http://geodata.mit.gov.it/datasets/parco_circolante_Abruzzo.csv)) : con il comando da shell `wc -l parco_circolante_Abruzzo.csv` leggo che è composto da 1170439 righe. Con una regione "piccola" come questa si va già oltre il limite di Calc, che è di 1048576 righe. Se voglio comunque visualizzare 1048576 righe di questo dataset, sul mio PC sono necessari circa **5 minuti**.

Con **VisiData** lancio `vd parco_circolante_Abruzzo.csv` e lo visualizzo in **1 secondo**. C'è il "trucco", che è invero una delle buone scelte del progettista: il file viene aperto in modo asincrono, caricandone subito una porzione navigabile. In basso a destra (vedi sotto) viene dato conto della percentuale di avanzamento che porterà alla piena apertura (meno di 15 secondi).



id	progressivo	tipo veicolo	destinazione	uso	co
1	27764	M	MOTOCICLO PER TRAS...	PROPRIO	TE...
2	48845	M	MOTOCICLO PER TRAS...	PROPRIO	VA...
3	65014	M	MOTOCICLO PER TRAS...	PROPRIO	AV...
4	70970	M	MOTOCICLO PER TRAS...	PROPRIO	CA...
5	97143	M	MOTOCICLO PER TRAS...	PROPRIO	LE...
6	106145	M	MOTOCICLO PER TRAS...	PROPRIO	L'...
7	140551	A	AUTOVETTURA PER TR...	PROPRIO	PE...
8	179506	M	MOTOCICLO PER TRAS...	PROPRIO	PO...
9	197102	M	MOTOCICLO PER TRAS...	PROPRIO	LA...
10	210700	M	MOTOCICLO PER TRAS...	PROPRIO	TE...
11	216934	A	AUTOCARAVAN	PROPRIO	PE...
12	241032	A	AUTOCARAVAN	PROPRIO	CO...
13	255318	A	AUTOCARRO PER TRAS...	PROPRIO	SC...
14	307257	A	AUTOVETTURA PER TR...	PROPRIO	AL...
15	310915	A	AUTOCARAVAN	PROPRIO	CA...
16	315875	M	MOTOCICLO PER TRAS...	PROPRIO	VA...
17	336492	A			RI...
18	343373	M	MOTOCICLO PER TRAS...	PROPRIO	AV...
19	345834	M	MOTOCICLO PER TRAS...	PROPRIO	OR...
20	420243	M	MOTOCICLO PER TRAS...	PROPRIO	PE...
21	435835	M	MOTOCICLO PER TRAS...	PROPRIO	AT...
22	481583	M	MOTOCICLO PER TRAS...	PROPRIO	TE...

parco\_circolante\_Abruzzo | saul.pw/VisiData v1.5 | opening pa 413790 25%...

Questo essere "subito pronto" lo rende uno strumento di grande comodità, che l'ha portato a essere uno dei miei quotidiani.

Ribadisco però ancora un volta che il "grande" e la valutazione del tempo dipendono dagli obiettivi che si hanno, dalle proprie conoscenze di base e dall'*hardware/software* che si ha a disposizione.

Per chiudere la tabella aperta e Visidata si pigia `q` sulla tastiera.

**NOTA BENE:** questo file con i dati sul "Parco Circolante dei veicoli" in Abruzzo, verrà usato in questa guida come dataset di base. Ne ho pertanto creato una copia di backup [qui](#).

### 1.0.0.2 Navigare tra i dati

Per muoversi tra le celle si usano le 4 frecce direzionali dalla tastiera o (come vim, da cui nell'uso da tastiera VisiData prende molta ispirazione) `h`, `j`, `k`, `l`.

Per muoversi in modo più esteso:

- `g` + freccia in basso oppure `gj`, per andare all'ultima riga;
- `g` + freccia in alto oppure `gk`, per andare alla prima riga;
- `g` + freccia a sinistra oppure `gh`, per andare alla colonna più a sinistra;
- `g` + freccia a destra oppure `gl`, per andare alla colonna più a destra;
- `PageDown` oppure `Control + Shift + f`, una pagina in giù;
- `PageUp` oppure `Control + Shift + b`, una pagina in alto.

**NOTA BENE:** `g` è il tasto per i comandi "globali".

Per muoversi tramite ricerca testuale, sfruttando le espressioni regolari:

- `/ + regex`, cerca in avanti nella colonna corrente;
- `? + regex`, cerca indietro nella colonna corrente;
- `g/ + regex`, cerca in avanti in tutte le colonne;
- `g? + regex`, cerca indietro in tutte le colonne;

Con `n` e `N` si va avanti e indietro rispetto agli elementi di *output* del risultato della ricerca.

Infine per saltare a una determinata riga o colonna, rispettivamente `zr` e `zc` seguito dal numero di riga e colonna (la numerazione inizia da zero).

## 1.0.1 I fogli

In VisiData ci sono tre tipi di fogli:

- i fogli sorgente, con i dati che si è scelto di aprire con VisiData;
- i fogli derivati, come quelli derivanti da filtraggio o la tabelle con le frequenze (vedi sotto);
- i metafogli, che descrivono e definiscono i dataset caricati (come quello che descrive ad esempio le colonne di una tabella, o il "foglio dei fogli").

Il "foglio dei fogli" si apre con `Shift + s` e fornisce l'elenco (e alcune informazioni correlate) di tutti i fogli aperti. Selezionandone uno e pigiando `Invio`, quest'ultimo si aprirà.

name	nRows	#	nCols#	nVisibleCols#	cursorDisplay	keyColNames	source	progressPct
sheets	3		8	8	sheets	name	[3]	0
parco_circolante_Pescara	270689		20	20	49		parco_circolante_Abruzzo.csv	0
parco_circolante_Abruzzo	1170438		20	20			parco_circolante_Abruzzo.csv	0

Per rinominarne uno dal "foglio dei fogli", basta selezionarlo, premere `e` e inserire il nuovo nome e poi dare `Invio`. O in alternativa, se si è davanti al foglio che si vuole rinominare, premere la `barra spaziatrice`, scrivere `rename-sheet`, digitare il nuovo nome e infine pigiare su `Invio`.

Il foglio correntemente aperto si chiude con `q`.



## 2 Lavorare sulle righe

La cosa più importante da comprendere è come selezionarle e deselectionarle, perché molte delle operazioni vengono eseguite proprio su quelle selezionate.

I tasti principali di selezione sono:

- **s**, per selezionare la riga corrente;
- **u**, per deselectionare la riga corrente;
- **t**, per invertire la selezione della riga corrente;
- **gs**, per selezionare tutte le righe;
- **gu**, per deselectionare tutte le righe;
- **gt**, per invertire i criteri di selezione per tutte le righe.

id	progressivo	tipo veicolo	destinazione	uso	co
1	27764	M	MOTOCICLO PER TRAS...	PROPRIO	TE...
2	48845	M	MOTOCICLO PER TRAS...	PROPRIO	VA...
3	65014	M	MOTOCICLO PER TRAS...	PROPRIO	AV...
4	70970	M	MOTOCICLO PER TRAS...	PROPRIO	CA...
5	97143	M	MOTOCICLO PER TRAS...	PROPRIO	LE...
6	106145	M	MOTOCICLO PER TRAS...	PROPRIO	L'...
7	140551	A	AUTOVETTURA PER TR...	PROPRIO	PE...
8	179506	M	MOTOCICLO PER TRAS...	PROPRIO	PO...
9	197102	M	MOTOCICLO PER TRAS...	PROPRIO	LA...
10	210700	M	MOTOCICLO PER TRAS...	PROPRIO	TE...
11	216934	A	AUTOCARAVAN	PROPRIO	PE...
12	241032	A	AUTOCARAVAN	PROPRIO	CO...
13	255318	A	AUTOCARRO PER TRAS...	PROPRIO	SC...
14	307257	A	AUTOVETTURA PER TR...	PROPRIO	AL...
15	310915	A	AUTOCARAVAN	PROPRIO	CA...
16	315875	M	MOTOCICLO PER TRAS...	PROPRIO	VA...
17	336492	A			RI...
18	343373	M	MOTOCICLO PER TRAS...	PROPRIO	AV...
19	345834	M	MOTOCICLO PER TRAS...	PROPRIO	OR...
20	420243	M	MOTOCICLO PER TRAS...	PROPRIO	PE...
21	435835	M	MOTOCICLO PER TRAS...	PROPRIO	AT...
22	481583	M	MOTOCICLO PER TRAS...	PROPRIO	TE...
parco_circolante_Abruzzo				t	1170438 rows

### 2.0.0.1 Selezione di righe tramite espressione regolare

- **| + termine da ricercare**, seleziona tutte le righe in cui c'è corrispondenza per la colonna corrente;
- **\ + termine da ricercare**, deselectiona tutte le righe in cui c'è corrispondenza per la colonna corrente;
- **g| + termine da ricercare**, seleziona tutte le righe in cui c'è corrispondenza per una qualsiasi colonna;
- **g\ + termine da ricercare**, deselectiona tutte le righe in cui c'è corrispondenza per una qualsiasi colonna;
- **,**, dato il valore della cella selezionata, seleziona tutte le righe in cui per la colonna corrente c'è corrispondenza;

- `g,` , seleziona tutte le righe uguali a quelle corrente.

Se ad esempio:

- ci si sposta nella colonna `provincia_residenza`
- poi si pigia `|` ;
- si digita `TERAMO` ;
- si preme `Invio` .

... si ottiene qualcosa come quella di sotto.

uso	comune_residenza	provincia_residenza	regione_residenza	eta
PROPRIO	TERAMO	TERAMO	ABRUZZO	52
PROPRIO	VASTO	CHIETI	ABRUZZO	43
PROPRIO	AVEZZANO	L'AQUILA	ABRUZZO	55
PROPRIO	CAPPADOCIA	L'AQUILA	ABRUZZO	56
PROPRIO	LETTOMANOPPELLO	PESCARA	ABRUZZO	49
PROPRIO	L'AQUILA	L'AQUILA	ABRUZZO	
PROPRIO	PESCARA	PESCARA	ABRUZZO	68
PROPRIO	POPOLI	PESCARA	ABRUZZO	65
PROPRIO	LANCIANO	CHIETI	ABRUZZO	48
PROPRIO	TERAMO	TERAMO	ABRUZZO	
PROPRIO	PESCARA	PESCARA	ABRUZZO	
PROPRIO	COLONNELLA	TERAMO	ABRUZZO	53
PROPRIO	SCAFA	PESCARA	ABRUZZO	
PROPRIO	ALANNO	PESCARA	ABRUZZO	57
PROPRIO	CAPISTRELLO	L'AQUILA	ABRUZZO	26
PROPRIO	VASTO	CHIETI	ABRUZZO	37
PROPRIO	RIVISONDOLI	L'AQUILA	ABRUZZO	78
PROPRIO	AVEZZANO	L'AQUILA	ABRUZZO	41
PROPRIO	ORSOGNA	CHIETI	ABRUZZO	43
PROPRIO	PESCARA	PESCARA	ABRUZZO	
PROPRIO	ATRI	TERAMO	ABRUZZO	74
PROPRIO	TERAMO	TERAMO	ABRUZZO	61
parco_circolante_Abruzzo			KEY_UP	1170438 rows

### 2.0.0.2 Selezione di righe tramite espressioni Python

È possibile fare una selezione di righe tramite un'espressione di Python ([qui](https://docs.python.org/3/tutorial/introduction.html) per approfondire sui concetti di base delle espressioni).

I comandi da tastiera sono:

- `z|` , per selezionare tutte le righe in cui l'espressione è valida;
- `z\` , per deselectare tutte le righe in cui l'espressione è valida.

Ad esempio seguendo questi step:

- `gu` , per deselectare tutto;
- `z|` , per attivare la selezione tramite espressione Python;

- scrivendo `provincia_residenza == "PESCARA" and sesso == "F"`;
- e pigiando su `Invio`.

... verranno selezionati tutti gli elementi in cui la provincia è "Pescara" ed il sesso è "F".

#### 2.0.0.3 Selezione di righe tramite espressioni Python e regex

- si digita `z|`;
- si scrive l'espressione, come ad esempio `re.search("^1", FieldB) and re.search("^2", FieldA)`, ovvero tutte le righe in cui il campo `FieldB` inizia per `1` e il campo `FieldA` per `2`.

#### 2.0.0.4 Selezionare un numero random di righe

- aprire una tabella;
- fare click sulla barra spaziatrice;
- scrivere `random-rows` (si può scrivere anche soltanto ad esempio `ran` e autocompletare con `TAB`) e premere `INVIO`;
- scrivere il numero di righe random che si vogliono selezionare.

#### 2.0.0.5 Spostare righe

Si fa con queste combinazioni:

- `Shift+j`, per spostare la riga corrente verso il basso;
- `Shift+k`, per spostare la riga corrente verso l'alto.

#### 2.0.0.6 Cancellare righe

- `d` per cancellare la riga corrente;
- `gd` per cancellare le righe selezionate.

#### 2.0.0.7 Modificare il contenuto

Questi i comandi di base:

- `e`, per modificare la cella corrente;
- `Enter`, per chiudere la modifica;
- `Control+c`, per cancellare la modifica;
- `Control+a`, per andare a inizio linea;
- `Control+e`, per andare a fine linea;
- `Control+k`, per cancellare il contenuto a partire dalla posizione del cursore.

## 3 Lavorare sulle colonne

Ogni colonna in VisiData ha 3 attributi principali:

- **Name**, il nome della colonna;
- **Width**, la larghezza a schermo della colonna (se impostata a 0, la colonna diventa invisibile);
- **Type**, per definire se contiene testi, numeri interi, numeri decimali, monete o date.

### 3.0.0.1 Il foglio delle colonne

In VisiData esistono i cosiddetti meta-fogli (vedi [sopra](#)): uno di questi è il foglio delle colonne, che elenca tutte le colonne e i relativi attributi di una tabella.

A partire dalla visualizzazione di una tabella, si apre con **Shift+c**.

name	width#	type	fmtstr	value
id	4			6
progressivo	13			106145
tipo_veicolo	14			M
destinazione	38			MOTOCICLO PER TRASPORTO P...
uso	9			PROPRIO
comune_residenza	18			L'AQUILA
provincia_residenza	21			L'AQUILA
regione_residenza	19			ABRUZZO
eta_intestatario	18	int	{:.0f}	
sezzo	7			
marca	18			HONDA
cilindrata	12			599.00
alimentazione	15			BENZ
kw	5			78
data_immatricolazione	23			2000-07-19 00:00:00
classe_euro	13			1
emissioni_co2	15			
massa_complessiva	19			387
revisione_in_regola	21			N
assicurazione_in_regola	25			N

parco\_circolante\_Abruzzo\_columns | KEY\_RESIZE\_ 20 columns

È una delle caratteristiche di VisiData che consente una rapida prima lettura del dato, che personalmente uso anche ad esempio per costruire brevi relazioni che contengono schema dati.

### 3.0.0.2 Impostare il tipo di campo

Come impostazione predefinita VisiData imposta a "testo" la tipologia di tutti i campi (che è quello che faccio di solito anche io alla prima apertura sia in un foglio elettronico che in Pandas).

Per definire il tipo di campo di una colonna della tabella che si sta visualizzando (cosa che consente di fare correttamente ad esempio operazioni di ordinamento, matematiche, ecc.), si deve navigare sino alla colonna su cui si vuole intervenire e poi pigiare:

- #, per definire un campo con numeri interi (0, -1, 5000000)
- %, per definire un campo con numeri decimali (0.5, -3.14, 23.45557)
- \$, per definire un campo con monete (\$4.99, €20)
- @, per definire un campo con date (2018-04-06, April 6, 2018, 04/06/2018)
- ~, per definire un campo di testo (qualsiasi cosa!)

### 3.0.0.2.1 IMPOSTARE NUMERO DI DECIMALI IN UN CAMPO FLOATING

Di default i campi floating sono renderizzati con due decimali (la formattazione è impostata come `%.02f`).

Per modificare la cosa si può aprire il foglio delle colonne (`SHIFT+c`) e modificare il valore di formattazione delle colonna `fmtstr` (vedi sotto).

name	width#	type	fmtstr	value	expr	aggregators
OC DENOMINAZIONE SLL	20			SLL NON ATTRIBUIBILE	0	#
OC_DENOMINAZIONE_SLL_frozen	20			SLL NON ATTRIBUIBILE	1	#
FINANZ_UE	15	float	%.02f	8045388.60	2	#
FINANZ_UE_FESR	18	float	%.02f	8045388.60	3	#
FINANZ_UE_FSE	17	float	%.02f		4	#
FINANZ_UE_FEASR	19				5	#

Per passare a tre decimali per un campo, basterà ad esempio modificare la cella della colonna di interesse da `%.02f` a `%.03f` (in edit di cella si va con il tasto `e`).

Se si vuole fare su più colonne:

- si apre sempre il foglio delle colonne;
- si selezionano le colonne di cui si vuole cambiare la formattazione;
- si va sulla colonna `fmtstr`;
- si va in modifica globale con `ge` e si inserisce il valore di formattazione desiderato;
- si preme `INVIO` per applicare la modifica.

### 3.0.0.2.2 IMPOSTARE FORMATO DATA PERSONALIZZATO

Se una cella contiene ad esempio la data `06/11/2020` (qui come 6 novembre 2020), e imposto la colonna come data con il tasto `@`, visidata la interpreta come `2020-06-11`, ovvero come 11 giugno.

Per impostarla correttamente bisogna impostare un formato data personalizzato:

- si va sulla colonna con le date;
- si digita `z@` e si preme `INVIO`;
- si imposta il formato - in questo caso `%d/%m/%Y` - e si preme `INVIO`.

In output si avrà `2020-11-06`.

### 3.0.0.3 Rinominare le colonne

Per farlo si naviga sino alla colonna che si vuole rinominare e poi:

- si pigia `^`;
- si digita il nome che si vuole assegnare;
- si pigia `Invio`.

### 3.0.0.4 Rinominare le colonne in blocco

Può essere utile ad esempio anteporre dei caratteri a tutte le colonne che iniziano per `t`.

Questi i passi:

- si apre il foglio colonne con `shift + c`;
- si va nella colonna con il nome delle colonne (è la prima);
- si digita `|`, si scrive `^t` e si pigia `INVIO` (saranno selezionate tutte le colonne che iniziano con il carattere `t`);
- si pigia `g*`, per attivare il trova e sostituisci via *regex*;
- si scrive `^(t.+)/TR_\1` e poi `INVIO`.

Tutte le colonne che iniziano per `t`, avranno un nuovo nome corrispondente al precedente, con `TR_` come prefisso. La colonna `tabacco`, diventerebbe ad esempio `TR_tabacco`.

### 3.0.0.5 Come espandere, ridurre e rimuovere colonne

Questi gli shortcut:

- `_`, per espandere la larghezza della colonna corrente, alla larghezza massima del testo contenuto nella stessa, tenendo conto delle righe visualizzate a schermo;
- `g_`, per espandere la larghezza di tutte le colonne, alla larghezza massima del testo contenuto, tenendo conto delle righe visualizzate a schermo;
- `z_ + n`, per impostare la larghezza della colonna corrente a `n` caratteri;
- `-`, per nascondere la colonna corrente (viene impostata a 0 la sua larghezza);
- `gv`, per scoprire tutte le colonne nascoste;
- `z-`, per ridurre la larghezza della colonna corrente alla sua metà.

### 3.0.0.6 Spostare una colonna

Ecco come farlo:

- `Shift+h`, sposta la colonna corrente di una posizione a sinistra;
- `Shift+l`, sposta la colonna corrente di una posizione a destra.

### 3.0.0.7 Modificare una colonna tramite un'espressione Python

A questa tabella si vuole assegnare al campo `FieldA` il valore di `FieldB`, tutte le volte che `FieldA` è vuoto.

--	--

FieldA	FieldB
1	
	3
1	12

Questa la procedura:

- `gs` per selezionare tutte le righe;
- `g=` per scrivere l'espressione Python;
- aggiungere come espressione `FieldB if FieldA==" " else FieldA`.

Ovvero (`FieldA`) è uguale `FieldB` se `FieldA` non contiene nulla, altrimenti è uguale a se stesso (rimane invariato).

### 3.0.0.8 Calcoli per colonna

È possibile avere restituito, per una colonna, il risultato di uno dei calcoli di aggregazione disponibili:

key	desc
<code>min</code>	minimum value
<code>max</code>	maximum value
<code>avg</code>	arithmetic mean of values
<code>mean</code>	arithmetic mean of values
<code>median</code>	median of values
<code>mode</code>	mode of values
<code>sum</code>	sum of values
<code>distinct</code>	distinct values
<code>count</code>	number of values
<code>list</code>	list of values
<code>stdev</code>	standard deviation of values

Ad esempio per avere la somma dei valori di una colonna con numeri interi:

- si imposta il tipo di colonna a numero intero (con `#`);
- si digita `z+` e poi `INVIO`;
- si digita `sum` e poi `INVIO`.

In basso, si leggerà il valore risultante.

**NOTA BENE:** se nessuna riga è selezionata si avrà il totale per colonna, altrimenti il calcolo sarà applicato per colonna alle sole righe selezionate.

### 3.0.0.9 Come definire una colonna chiave

Per ogni foglio è possibile definire una o più colonne chiave. Hanno due funzioni principali:

- rimangono fissate sul lato sinistro, quando si naviga orizzontalmente sulla tabella;
- hanno una funzione speciale quando si fanno certe operazioni (come quelle di *JOIN*, per alcuni conteggi, ecc.).

Per definire una colonna chiave (e vice-versa) si naviga sino alla colonna di interesse e si pigia **!**.

### 3.0.0.10 Manipolare le colonne dal foglio delle colonne

Il foglio delle colonne oltre a dare una visione di insieme, consente di fare delle rapide modifiche in blocco. È possibile spostare le colonne con **Shift+j** e **Shift+k**, rinominarle con **e**, impostare il tipo per tutte quelle selezionate, impostare la larghezza per tutte quelle selezionate.

Ad esempio a partire dal Parco Circolante dei veicoli dell'Abruzzo, il foglio colonne (**Shift+c**) appare in questo modo.

name	width	#	type	fmtstr	value	exp>
id	4				1	0
progressivo	13				27764	1
tipo_veicolo	14				M	2
destinazione	38				MOTOCICLO PER TRAS...	3
uso	9				PROPRIO	4
comune_residenza	18				TERAMO	5
provincia_residenza	21				TERAMO	6
regione_residenza	19				ABRUZZO	7
eta_intestatario	18				52	8
Sesso	7				M	9
marca	18				SUZUKI	10
cilindrata	12				385.00	11
alimentazione	15				BENZ	12
kw	5				24	13
data_immatricolazi...	23				2005-12-21 00:00:00	14
classe_euro	13				2	15
emissioni_co2	15					16
massa_complessiva	19				395	17
revisione_in_regola	21				S	18
assicurazione_in_r...	25				S	19
parco_circolante_Abruzzo_columns						C 20 columns

Se si vuole impostare a "numero intero" le colonne `progressivo` e `eta_intestatario`, bisognerà prima selezionarle con **s**.



name	width#	type	fmtstr	value
id	4			1
progressivo	13			27764
tipo_veicolo	14			M
destinazione	38			MOTOCICLO PER TRASPORTO P...
uso	9			PROPRIO
comune_residenza	18			TERAMO
provincia_residenza	21			TERAMO
regione_residenza	19			ABRUZZO
eta_intestatario	18			52
Sesso	7			M
marca	18			SUZUKI
cilindrata	12			385.00
alimentazione	15			BENZ
kw	5			24
data_immatricolazione	23			2005-12-21 00:00:00
classe_euro	13			2
emissioni_co2	15			
massa_complessiva	19			395
revisione_in_regola	21			S
assicurazione_in_regola	25			S

parco\_circolante\_Abruzzo\_columns| KEY\_DOWN 20 columns

Poi premere g#.

name	width#	type	fmtstr	value
id	4			1
progressivo	13	int	{:.0f}	27764
tipo_veicolo	14			M
destinazione	38			MOTOCICLO PER TRASPORTO P...
uso	9			PROPRIO
comune_residenza	18			TERAMO
provincia_residenza	21			TERAMO
regione_residenza	19			ABRUZZO
eta_intestatario	18	int	{:.0f}	52
Sesso	7			M
marca	18			SUZUKI
cilindrata	12			385.00
alimentazione	15			BENZ
kw	5			24
data_immatricolazione	23			2005-12-21 00:00:00
classe_euro	13			2
emissioni_co2	15			
massa_complessiva	19			395
revisione_in_regola	21			S
assicurazione_in_regola	25			S

parco\_circolante\_Abruzzo\_columns| g# 20 columns

### 3.0.0.11 Creare una nuova colonna

Sono tre le modalità più comode:

- come risultato di un'espressione Python;
- come risultato di un comando di *split*;
- come risultato di estrazione di una porzione di testo da una colonna.

#### 3.0.0.11.1 NUOVA COLONNA TRAMITE ESPRESSIONE

Ad esempio potrebbe essere utile creare una colonna vero/falso, per tutte le auto del Parco Circolante dei veicoli dell'Abruzzo, con cilindrata maggiore ai 643 (è un numero a caso). Per farlo:

- navigare sino alla colonna `cilindrata`;
- premere `%` per impostarla come numero decimale;
- pigiare su `=`;
- nel *prompt* in basso a sinistra scrivere `cilindrata > 643`;
- premere `Invio`.

In output qualcosa come quella di sotto, con la nuova colonna risultato di un'espressione Python.

eta	intestatario	sezzo	marca	cilindrata%	cilindrata > 643
52		M	SUZUKI	385.00	False
43		M		595.00	False
55		M		124.00	False
56		M	APRILIA	151.00	False
49		M	MBK	124.00	False
			HONDA	599.00	False
68		M	RENAULT	1149.00	True
65		M		643.00	False
48		M		749.00	True
			HONDA	279.00	False
			FIAT - INNOCENTI	2800.00	True
53		M	FORD	2402.00	True
				13798.00	True
57		M	TOYOTA	1364.00	True
26		M	FIAT - INNOCENTI	1905.00	True
37		M		595.00	False

#### 3.0.0.11.2 NUOVE COLONNE COME RISULTATO DI SPLITTING

Si possono creare nuove colonne, come risultato di una suddivisione di una già esistente. Ad esempio nel campo `data_immatricolazione` è contenuta sia la data che l'orario, separati da spazio (i.e. `1995-04-03 00:00:00`).

Se si vuole suddividerla in due colonne (data e orario), si può procedere in questo modo:

- si naviga sino alla colonna `data_immatricolazione`;
- si preme `:`;
- si sceglie la stringa di testo (definibile anche come espressione regolare) da usare come separatore e si scrive in basso a sinistra nel *prompt*. In questo caso è lo spazio;

- si digita quindi lo spazio e si dà `Invio`.

Come output si avrà qualcosa come quella di sotto: due colonne frutto della divisione in più colonne, sfruttando come separatore lo spazio.

<code>data_immatricolazione</code>	<code>data_immatricolazione_re0</code>	<code>data_immatricolazione_re1</code>
2005-12-21 00:00:00	2005-12-21	00:00:00
1995-04-03 00:00:00	1995-04-03	00:00:00
1986-01-01 00:00:00	1986-01-01	00:00:00
2001-11-09 00:00:00	2001-11-09	00:00:00
2000-11-10 00:00:00	2000-11-10	00:00:00
2000-07-19 00:00:00	2000-07-19	00:00:00
1999-08-11 00:00:00	1999-08-11	00:00:00
1987-05-06 00:00:00	1987-05-06	00:00:00
1979-01-01 00:00:00	1979-01-01	00:00:00
2015-07-03 00:00:00	2015-07-03	00:00:00
2000-02-23 00:00:00	2000-02-23	00:00:00
2006-03-28 00:00:00	2006-03-28	00:00:00
1981-01-01 00:00:00	1981-01-01	00:00:00
2006-04-27 00:00:00	2006-04-27	00:00:00
2001-04-24 00:00:00	2001-04-24	00:00:00

### 3.0.0.11.3 NUOVE COLONNE COME RISULTATO DI CATTURA E ESTRAZIONE

Per farlo ci vogliono delle conoscenze di base sulle espressioni regolari (che sono una conoscenza propedeutica per chiunque lavori con i dati).

Se si vuole ad esempio estrarre l'anno dalla celle che contiene il valore `1995-04-03 00:00:00`, basterà definire l'espressione regolare che estrae a partire dall'inizio della cella i primi 4 numeri. Che in termini di espressione regolare è `^[0-9]{4}`.

Per creare una nuova colonna a partire dalla definizione di un'espressione regolare da usare come "estrattore", si procede in questo modo:

- si naviga sulla colonna `data_immatricolazione`;
- si pigia `;`;
- si scrive nel prompt in basso a sinistra `^[0-9]{4}`. Sono state inserite le parentesi tonde, perché la cattura si fa a partire dalla definizione di un gruppo;
- si pigia `Invio`.

E si ottiene qualcosa come quella di sotto:

alimentazione	kw	data immatricolazione	data immatricolazione_re0
BENZ	24	2005-12-21 00:00:00	2005
BENZ	33	1995-04-03 00:00:00	1995
B/OLIO	17	1986-01-01 00:00:00	1986
BENZ	10	2001-11-09 00:00:00	2001
BENZ	8	2000-11-10 00:00:00	2000
BENZ	78	2000-07-19 00:00:00	2000
BENZ	43	1999-08-11 00:00:00	1999
BENZ	35	1987-05-06 00:00:00	1987
BENZ	38	1979-01-01 00:00:00	1979
BENZ	19	2015-07-03 00:00:00	2015
GASOL	90	2000-02-23 00:00:00	2000
GASOL	101	2006-03-28 00:00:00	2006
GASOL	224	1981-01-01 00:00:00	1981
GASOL	66	2006-04-27 00:00:00	2006

### 3.0.0.12 Trasformare una colonna tramite espressione regolare (trova e sostituisci)

Su ogni colonna è possibile fare delle trasformazioni basate su espressioni regolari. Se ad esempio si vogliono rimuovere dalle celle della colonna `data_immatricolazione` le informazioni sull'orario e passare quindi da `1995-04-03 00:00:00` a `1995-04-03` si può procedere in questo modo:

- si selezionano le righe su cui si vuole fare questa trasformazione (se sono tutte il comando è `gs`);
- si va nella colonna `data_immatricolazione`;
- si pigia su `g*`;
- si scrive nel prompt `.*$ /`. Il simbolo `/` in VisiData separa ciò che si cerca, da ciò con cui si vuole sostituire. Qui si cerca uno spazio ( ), seguito da qualsiasi carattere sino a fine cella ( `.*$` ), e si vuole sostituire con nulla;
- si pigia `Invio` (ci vorrà un po' di tempo, sono più di un milione di righe).

Il risultato sarà qualcosa come quella di sotto:

cilindrata	alimentazione	kw	data immatricolazione
385.00	BENZ	24	2005-12-21
595.00	BENZ	33	1995-04-03
124.00	B/OLIO	17	1986-01-01
151.00	BENZ	10	2001-11-09
124.00	BENZ	8	2000-11-10
599.00	BENZ	78	2000-07-19
1149.00	BENZ	43	1999-08-11
643.00	BENZ	35	1987-05-06
749.00	BENZ	38	1979-01-01
279.00	BENZ	19	2015-07-03
2800.00	GASOL	90	2000-02-23
2402.00	GASOL	101	2006-03-28
13798.00	GASOL	224	1981-01-01
1364.00	GASOL	66	2006-04-27
1905.00	GASOL	66	2001-04-24

Questo è un "trova e sostituisci" per colonna.

### 3.0.0.13 Il foglio delle frequenze

Questa è una delle funzioni che uso di più ed è un'altra gran bell'idea. A partire da una colonna (o dalla combinazione di più d'una) è possibile - con `Shift+f` - aprire un foglio derivato, che contiene dei dati relativi al conteggio per valori distinti di quanto contenuto in quella colonna.

Se per esempio a partire dal Parco Circolante dei veicoli dell'Abruzzo, si volesse avere un'idea sulla distribuzione per "marca", basta navigare in vista tabella sino a quella colonna e poi pigiare `Shift+f` e avere qualcosa come quella di sotto.

marca	count #	percent%	histogram
FIAT - INNOCENTI	301849	25.79	*****
FORD	65677	5.61	*****
	65070	5.56	*****
VOLKSWAGEN	59586	5.09	*****
OPEL	56495	4.83	*****
LANCIA - AUTOBIANCHI	51699	4.42	*****
RENAULT	41891	3.58	*****
CITROEN	41208	3.52	*****
MERCEDES	36572	3.12	*****
TOYOTA	34850	2.98	*****
ALFA ROMEO	34158	2.92	*****
PEUGEOT	33733	2.88	*****
IVECO - FIAT	28472	2.43	****
PIAGGIO	26639	2.28	****
NISSAN	25953	2.22	****
BMW	24364	2.08	****

I dati sono ordinati per conteggio ed emerge subito come una famosa marca nazionale da sola faccia  $\frac{1}{4}$  del campione. Così come emerge che c'è un 5% di elementi in cui la marca non è definita e anche questa è un'informazione che ci potrebbe consentire da subito di prendere delle decisioni.

Molto comoda anche la colonna `histogram` che da visivamente in modo molto diretto un'idea del rapporto tra i numeri descritti in tabella.

## 4 Lavorare sulle celle

### 4.0.0.1 Fare un trova e sostituisci globale

Si parte dal selezionare le righe su cui si vuole eseguire l'operazione (se sono tutte il comando è `gs`).

Digitare `gz*` e poi scrivere `stringa da trovare / stringa da sostituire` e dare `INVIO`.

Se ad esempio volessi sostituire la stringa `pal` con `mis`, dovrei scrivere `pal/mis`.

È possibile usare anche le espressioni regolari.

## 5 Ordinare e filtrare

### 5.0.0.1 Ordinare righe

I tasti `[ ]` ordinano i dati rispettivamente in modo crescente e decrescente, a partire dalla colonna attiva.

### 5.0.0.2 Filtrare righe

È possibile estrarre in vari modi un campione delle righe della tabella visualizzata. Quello di base è a partire dalla selezione delle righe (in uno dei modi visti sopra) e poi pigiare `"`. Ad esempio si naviga sino alla colonna `marca`, si preme `|`, si scrive `chevr`, si pigia `Invio` (vengono selezionati tutti i record associati a `CHEVROLET`) e infine si preme `"`. Si avrà una tabella filtrata, con gli 8814 record relativi a questa marca per l'Abruzzo.

Oppure posso farlo a partire da un "foglio delle frequenze" (vedi [qui](#)):

- si naviga sino alla colonna `marca`;
- si pigia `Shift+f`, che mi dà in *output* il conteggio dei valori distinti per marca (sono più di un milione di record, ci vorrà un po' di tempo);
- si scorre sino alla riga che contiene la marca che si vuole filtrare;

marca	count #	percent %	histogram
FIAT - INNOCENTI	301849	25.79	*****
FORD	65677	5.61	*****
	65070	5.56	*****
VOLKSWAGEN	59586	5.09	*****
OPEL	56495	4.83	*****
LANCIA - AUTOBIANC...	51699	4.42	*****
RENAULT	41891	3.58	*****
CITROEN	41208	3.52	*****

- si pigia `Invio` e si avrà una tabella filtrata, con i record relativi a quella marca per l'Abruzzo.

RENAULT	7	140551	A	AUTOVETTURA PE...
RENAULT	130	2342318	A	AUTOCARRO PER ...
RENAULT	197	3444450	A	AUTOVETTURA PE...
RENAULT	198	3474466	A	AUTOVETTURA PE...
RENAULT	238	4287789	A	AUTOVETTURA PE...
RENAULT	246	4411276	A	AUTOCARRO PER ...
RENAULT	301	5819148	A	AUTOCARAVAN
RENAULT	354	6534799	A	AUTOCARRO PER ...
RENAULT	364	6576187	A	AUTOVETTURA PE...
RENAULT	414	6759292	A	AUTOCARRO PER ...
RENAULT	431	6823682	A	AUTOVETTURA PE...
RENAULT	451	6906594	A	AUTOVETTURA PE...
RENAULT	488	7047830	A	AUTOVETTURA PE...
RENAULT	716	12296570	A	AUTOCARRO PER ...
RENAULT	762	13170016	A	AUTOVETTURA PE...
RENAULT	811	14271772	A	AUTOVETTURA PE...
RENAULT	850	15816058	A	AUTOVETTURA PE...
RENAULT	864	16125934	A	AUTOVETTURA PE...
RENAULT	878	16311903	A	AUTOVETTURA PE...
RENAULT	916	16526628	A	AUTOVETTURA PE...
RENAULT	917	16550144	A	AUTOVETTURA PE...
RENAULT	921	16558985	A	AUTOVETTURA PE...

parco\_circolante\_Abruzzo\_RENAULT | KEY\_RESIZE\_ 41891 rows

Dal foglio delle frequenze è possibile filtrare su più di un elemento:

- si selezionano ad esempio tre marche, pigiando **s** per ognuna;
- si chiude il foglio delle frequenze con **q** e si torna al foglio dati;
- si pigia **"** e si ottiene una tabella filtrata con i dati relativi alle sole tre marche selezionate prima.

#### 5.0.0.2.1 FILTRO TRAMITE ESPRESSIONE PYTHON

È possibile **filtrare** righe tramite un' **espressione Python** ([qui](https://docs.python.org/3/reference/expressions.html) [la](https://docs.python.org/3/reference/expressions.html) documentazione ufficiale e [qui](https://realpython.com/python-operators-expressions/) [una](https://realpython.com/python-operators-expressions/) guida che consiglio). Se ad esempio dalla tabella di sotto si volessero soltanto le province con più di 350.000 abitanti

Provincia	Popolazione
L'AQUILA	300404
TERAMO	308284
PESCARA	319388
CHIETI	387120

basterebbe:

- per prima cosa impostare il campo **Popolazione** come numerico, pigiando **#**;
- pigiare **z** per attivare il filtro tramite espressione Python;
- scrivere l'espressione basata sulla colonna, che qui è **Popolazione>350000** e poi premere **INVIO**.

In output si avrà la selezione di tutte le righe che risolvono l'espressione scritta.



Provincia	Popolazione #
L'AQUILA	300404
TERAMO	308284
PESCARA	319388
CHIETI	387120

Note:

- nel filtro per espressione c'è l'autocompletamento del nome colonna, quindi se si scrive ad esempio `Pop` e poi si fa click su `TAB`, viene restituito a schermo `Popolazione`;
- è possibile scrivere espressioni complesse, che fanno riferimento a più colonne.

#### 5.0.0.2.2 FILTRO TRAMITE ESPRESSIONE PYTHON SU CAMPO DATETIME

È possibile filtrare righe tramite un'espressione Python a partire da campi `datetime`, dopo averli impostare come `data`.

Usando [questo](https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-andamento-nazionale/dpc-covid19-ita-andamento-nazionale.csv) (<https://raw.githubusercontent.com/pcm-dpc/COVID-19/master/dati-andamento-nazionale/dpc-covid19-ita-andamento-nazionale.csv>) file CSV (by PCM-DPC (<https://github.com/pcm-dpc/COVID-19>)), se ad esempio dalla tabella di sotto si volessero soltanto i record con giorno 29:

data	stato	ricoverati
2020-02-24	ITA	101
2020-02-25	ITA	114
2020-02-26	ITA	128
2020-02-27	ITA	248
2020-02-28	ITA	345
2020-02-29	ITA	401
2020-03-01	ITA	639

basterebbe:

- per prima cosa impostare il campo `data` come `data`, pigiando `@`;
- pigiare `z` per attivare il filtro tramite espressione Python;
- scrivere l'espressione basata sulla colonna, che qui è `data.day == 29` e poi premere `INVIO`.

In output si avrà la selezione di tutte le righe che risolvono l'espressione scritta.

data	stato	ricoverati
2020-02-24	ITA	101
2020-02-25	ITA	114
2020-02-26	ITA	128
2020-02-27	ITA	248
2020-02-28	ITA	345
2020-02-29	ITA	401
2020-03-01	ITA	639

Osservazioni: oltre a `day` è possibile usare: `month`, `year` e `hour`

Se si vogliono ad esempio selezionare soltanto righe con date nel futuro, l'espressione sarà `data > datetime.datetime.now()`.

#### 5.0.0.2.3 FILTRI TRAMITE ESPRESSIONI PYTHON, BASATI SU PIÙ COLONNE

Si vogliono ad esempio selezionare tutte le righe in cui il domicilio non è né a Palermo, né a Ragusa, ma la cui residenza è in una di queste due città.

A partire ad esempio da:

domicilio	residenza
PA	PA
RG	RG
TO	TO
VE	PA

Questa la procedura:

- pigiare `z|` per selezionare tramite espressione Python;
- scrivere `re.search("^(?!PA|RG).*", domicilio)` and `re.search("(PA|RG)", residenza)`;
- premere `INVIO`.

La prima è una speciale condizione di `regex`, che cerca al negativo.

Sarà selezionata soltanto la riga seguente:

domicilio	residenza
VE	PA

## 6 Riepilogo dei dati

Uno dei modi per avere un riepilogo è usare il foglio delle frequenze (vedi [qui](#)).

È possibile generare un foglio delle frequenze anche basato su più colonne. Per farlo si impostano come colonne chiave ([qui](#) come fare) quelle che si vogliono trasformare in "foglio delle frequenze", e poi si pigia `g+Shift+f`. Qui sotto ad esempio un riepilogo per le coppie univoche di `destinazione/alimentazione`.

destinazione	alimentazione	count #	percent%
AUTOVETTURA PER TRASPORTO DI PERSONE	GASOL	416652	35.60
AUTOVETTURA PER TRASPORTO DI PERSONE	BENZ	355012	30.33
MOTOCICLO PER TRASPORTO PERSONE	BENZ	121903	10.42
AUTOCARRO PER TRASPORTO DI COSE	GASOL	99977	8.54
AUTOVETTURA PER TRASPORTO DI PERSONE	B/GPL	70801	6.05
AUTOVETTURA PER TRASPORTO DI PERSONE	B/MET	26624	2.27
MOTOCICLO PER TRASPORTO PERSONE	B/OLIO	11552	0.99
MOTOCICLO PER TRASPORTO PERSONE	MISCELA	8812	0.75
AUTOVEIC.TRASP.PROMISCUO PERSONE/COSE	GASOL	8695	0.74
AUTOVEICOLO PER TRASPORTO SPECIFICO	GASOL	8074	0.69
AUTOVEICOLO PER USO SPECIALE	GASOL	5358	0.46
AUTOCARAVAN	GASOL	4460	0.38
AUTOVEIC.TRASP.PROMISCUO PERSONE/COSE	BENZ	4357	0.37
TRATTORE PER SEMIRIMORCHIO	GASOL	3975	0.34
QUADRICICLO PER TRASP. DI PERSONE	BENZ	3270	0.28
AUTOCARRO PER TRASPORTO DI COSE	BENZ	3225	0.28
AUTOBUS PER TRASPORTO DI PERSONE	GASOL	3036	0.26
TRICICLO PER TRASPORTO COSE	B/OLIO	2243	0.19

### 6.0.0.1 Aggiunta di aggregatori

I fogli di frequenza, oltre al conteggio per valori distinti, possono contenere altri calcoli. Gli "aggregatori" possibili sono `min`, `max`, `avg / mean`, `median`, `q3/q4/q5/q10` (terzili/quartili/quintili/decili), `sum`, `distinct`, `count` e `keymax`.

Ad esempio per ogni provincia si può avere restituito la distribuzione di età per quartile e scoprire che il 50% del campione è compreso nella fascia di età tra i 18 e circa i 55 anni.

provincia_residenza	count #	p25 eta intestatario#	p50 eta intestatario#	p75 eta intestatario#
CHIETI	142995	44	54	65
TERAMO	119454	45	54	65
PESCARA	111916	44	53	64
L'AQUILA	106634	44	54	65

Per aggiungere l'aggregatore `q4` al foglio delle frequenze, bisogna seguire questi passi:

- navigare sino alla colonna `eta_intestatario` e impostarla a numero intero con `#`;
- premere `+` e scrivere (in basso a sinistra) `q4` (per il calcolo dei quartili, quindi al 25, 50 e 75 percento);
- navigare sino alla colonna `provincia_residenza` e pigiare `Shift+f`.

Inizierà da subito il calcolo (che non è immediato, sono più di un milione di righe) e alla fine si avrà qualcosa come l'immagine di sopra (non è fico VisiData?).

### 6.0.0.2 Il riepilogo globale

Per avere una visione a "volo d'uccello" sull'intera tabella esiste il comando `Shift+i`, che restituisce un riepilogo statistico per tutte le colonne.

Se prima di lanciarlo si definiscono correttamente i campi (ad esempio i numerici, come numerici), verranno eseguiti correttamente anche i calcoli per `min`, `max`, `median`, `mean`, `stdev`.

column	errors#	nulls#	distinct#	mode	~	min~	max	~	median	~	mean	%	stdev	%
id	0	0	1170438		1	1	1170438		585219.5		585219.50		337876.49	
progressivo	0	0	1170438		1	1	52165186		22894754.5		24994269.45		15139735.31	
tipo_veicolo	0	0	2	A										
destinazione	0	359	25	AUTOVETTURA PER T...										
uso	0	359	8	PROPRIO										
comune_residenza	0	70	308	PESCARA										
provincia_residenza	0	0	4	CHIETI										
regione_residenza	0	0	1	ABRUZZO										
eta_intestataro	0	109003	118	53	1	1	2017		53		53.91		15.85	
Sesso	0	115868	3	M										
marca	0	65070	136	FIAT - INNOCENTI										
cilindrata	0	875	2107	1242.0	0.0	0.0	29988.0		1364.0		1571.40		1436.96	
alimentazione	0	690	14	GASOL										
kw	0	692	390	51	0	0	720		55.0		64.59		40.82	
data_immatricolazione	0	3032	13803	1990-01-01 00:00:...										
classe_euro	0	2	8	4	0	0	6		3.0		3.11		1.75	
emissioni_co2	0	441765	420	119.0	0.0	0.0	999.0		134.0		137.23		33.02	
massa_complessiva	0	3662	2501	3500	0	0	99999		1550.0		1895.81		2611.21	
revisione_in_regola	0	0	2	S										
assicurazione_in_regola	0	0	2	S										

Questa è un'altra *feature* di grande comodità (presente in molte applicazioni e ambienti per analisi dati), che verrà usata molto da chi lavorerà con VisiData.

## 7 Salvare un foglio

Per salvare un foglio si preme `CTRL+s`, poi si scrive il nome del file con estensione ( `nomeEsempio.csv` ) e infine si pigia `Invio`.

## 8 Fare il redirect dell'output verso lo stdout

### Il comando

```
vd input.csv -b --save-filetype json | jq .
```

invierà ad esempio a `jq` un *output* `JSON` a partire dal CSV di *input*.

Se ad esempio si vuole passare l'output a `grep`:

```
vd input.csv -b --save-filetype tsv 2> /dev/null | grep 'a'
```

`2> /dev/null` per non avere `stderr` a schermo.

## 9 Fare JOIN tra tabelle

Se ad esempio si volesse calcolare il rapporto tra numero di mezzi e popolazione, sarebbe utile fare un JOIN con una tabella con i dati della popolazione residente.

I dati per provincia sono quelli di sotto e sono stati salvati in un file TSV denominato `popolazioneAbruzzo.tsv` (fonte ISTAT (<http://dati.istat.it/Index.aspx?QueryId=18545#>)).

Provincia	Popolazione
L'AQUILA	300404
TERAMO	308284
PESCARA	319388
CHIETI	387120

Si può fare in questo modo:

- si parte da `vd parco_circolante_Abruzzo.csv`;
- si va sulla colonna `provincia_residenza`;
- si calcola il foglio delle frequenze pigiando `Shift+f` e si ottiene

provincia_residenza	count #	percent%	histogram
CHIETI	350535	29.95	*****
TERAMO	274801	23.48	*****
L'AQUILA	274413	23.45	*****
PESCARA	270689	23.13	*****

- si rinomina questo foglio frequenze premendo la barra spaziatrice, scrivendo il comando `rename-sheet` e dando poi `Invio`. E poi inserendo il nome nuovo (ad esempio `mezziProvincia`);
- si apre la tabella con i dati sulla popolazione per provincia, pigiando `o`, scrivendo `popolazioneAbruzzo.tsv` (che è il nome del file) e pigiando `Invio` (NOTA BENE si può scrivere anche soltanto `popo` e poi pigiare `TAB` e il nome del file verrà autocompletato). Si otterrà

Provincia	Popolazione
L'AQUILA	300404
TERAMO	308284
PESCARA	319388
CHIETI	387120

- si va nella colonna `Provincia` e si preme `!` per impostarla come colonna chiave. Il JOIN in VisiData viene fatto tra colonne chiave;
- si apre il "foglio dei fogli" con `Shift+s` e si visualizzerà qualcosa come;

name	nRows	#	nCols	#	nVisibleCols	#
sheets	4		8		8	
popolazioneAbruzzo	4		2		2	
mezziProvincia	4		4		4	
parco_circolante_Abruzzo	1170438		20		20	

- si va nella riga che contiene lo sheet `popolazioneAbruzzo` e si pigia `INVIO`;
- si va di nuovo nella colonna `Provincia` e si preme `!` per impostarla come colonna chiave;
- si apre il "foglio dei fogli" con `Shift+s`;
- si selezionano con `s` le due tabelle `mezziProvincia` e `popolazioneAbruzzo`;
- si pigia `&` che è il comando di JOIN e in basso a sinistra ci viene chiesto quale tipo si vuole applicare (nota bene: con VisiData >2.0 si deve premere `CTRL + x` e scegliere il tipo di JOIN);

name	nRows	#	nCols	#	nVisibleCols	#	cursorDisp>
sheets	4		8		8		parco_circ...
popolazioneAbruzzo	4		2		2		L'AQUILA
mezziProvincia	4		4		4		CHIETI
parco_circolante_Abruzzo	1170438		20		20		TERAMO

inner/outer/full/diff/append/extend:  & 4 sheets

- si scrive `inner` e si pigia `Invio` e si ottiene una tabella che contiene sia il numero di mezzi per provincia, che il numero di abitanti.

Provincia	Popolazione	count #	percent%
L'AQUILA	300404	274413	23.45
TERAMO	308284	274801	23.48
PESCARA	319388	270689	23.13
CHIETI	387120	350535	29.95

Da questa tabella a questo punto, semplificando un po', si potrebbe rapidamente calcolare il numero macchine per persona, in questo modo:

- si va nella colonna `Popolazione` e si preme `#` per impostarla come numero intero;
- si pigia `=` (per creare una nuova colonna con valori frutto di un'espressione);
- si scrive poi nel `prompt` `count/Popolazione` (c'è l'autocompletamento con `TAB` dei nomi delle colonne);
- e si ottiene un risultato come quello di sotto.

Provincia	Popolazione#	count #	count/Popolazione%	percent%
PESCARA	319388	270689	0.85	23.13
TERAMO	308284	274801	0.89	23.48
L'AQUILA	300404	274413	0.91	23.45
CHIETI	387120	350535	0.91	29.95



## 10 Concatenare tabelle

La concatenazione tra due o più tabelle con la stessa struttura si attiva tramite `&`. Ad esempio, a partire dai due file `inputFileOne.csv` e `inputFileTwo.csv`, presenti [qui \(https://github.com/ondata/guidaVisiData/tree/master/dati\)](https://github.com/ondata/guidaVisiData/tree/master/dati) si procede in questo modo:

- si apre la shell e si va nella cartella che contiene i file da concatenare;
- si scrive `vd` e si preme `Invio`;
- si seleziona la prima voce `DirSheet for the current directory` e si preme `invio`;
- si selezionano con `s` i due file;
- si preme `g Invio` per aprirli entrambi;
- si apre il foglio dei fogli con `Shift+s`;
- si selezionano `inputFileOne.csv` e `inputFileTwo.csv` con `s`;
- si preme `&` per attivare la concatenazione;
- si preme `Ctrl+X` per il menu;
- si seleziona `append` e poi si preme `Invio`.

Si avrà a video l'unione dei due file di input (in [questo video \(http://youtu.be/NJ8FBYI-WGE?hd=1\)](http://youtu.be/NJ8FBYI-WGE?hd=1) una replica di quanto descritto, per versioni `vd < 2`).

# 11 Tabelle Pivot

Ecco ad esempio come creare quella che da conto del numero di mezzi per marca, suddivisi per provincia:

- si va nella colonna `marca` e si imposta come colonna chiave, premendo `!`;
- si va poi nella colonna `provincia_residenza` e si pigia `Shift+w`;
- si ottiene la tabella pivot desiderata.

marca	TERAMO#	CHIETI#	L'AQUILA#	PESCARA#	Total_count#
FIAT - INNOCENTI	72980	97824	67004	64041	301849
FORD	15633	18675	16772	14597	65677
	13858	21133	15392	14687	65070
VOLKSWAGEN	14142	17516	16027	11901	59586
OPEL	13854	15627	12737	14277	56495
LANCIA - AUTOBIANCHI	12112	16153	10927	12507	51699
RENAULT	10783	12165	9787	9156	41891
CITROEN	9036	11234	10415	10523	41208
MERCEDES	9685	9855	8595	8437	36572
TOYOTA	7868	9135	8657	9190	34850
ALFA ROMEO	7318	10895	8983	6962	34158
PEUGEOT	7598	10566	8038	7531	33733
IVECO - FIAT	7716	8552	6776	5428	28472
PIAGGIO	5975	9207	3059	8398	26639
NISSAN	5608	6029	8086	6230	25953
BMW	6284	6580	5789	5711	24364
AUDI	5158	7090	6701	4899	23848
HONDA	5043	7495	4959	6322	23819
SUZUKI	4925	4541	6253	4447	20166
YAMAHA	3586	5667	2611	5049	16913
HYUNDAI	2872	4039	2442	4448	13801

Se invece del conteggio delle occorrenze si volesse (a partire da un altro campo) calcolare un altro dato aggregato - come ad esempio l'età media per marca e provincia - si può procedere in questo modo:

- si va nella colonna `eta_intestatario` e si imposta a numero intero, pigiando su `#`;
- si preme `+`, poi in basso a sinistra nel `prompt` si scrive `avg` e infine si dà `Invio`; si va di nuovo nella colonna `provincia_residenza` e si pigia `Shift+w`.

In output, per ogni provincia, si avrà quindi la media desiderata.

marca	eta_intestataro_avg_TERAMO%	eta_intestataro_avg_CHIETI%	eta_intestataro
SUZUKI	52.91	52.52	
	58.19	58.42	
APRILIA	49.30	49.44	
MBK	50.09	49.59	
HONDA	50.35	50.89	
RENAULT	53.51	54.74	
FIAT - INNOCENTI	56.27	56.18	
FORD	53.81	53.55	
TOYOTA	53.88	53.94	
DUCATI	42.93	42.40	
BMW	51.00	50.43	
PEUGEOT	53.44	53.22	
YAMAHA	47.90	48.30	
MERCEDES	53.25	52.86	
KYMCO	50.60	51.43	
ALFA ROMEO	52.62	51.37	
HYUNDAI	54.51	54.75	
NISSAN	52.70	52.64	
BETA	48.25	46.87	
AUDI	50.54	49.63	

## 12 Creare un grafico di dispersione (scatter plot)

VisiData può stampare a schermo anche dei grafici XY. Non è un suo punto di forza, ma può essere utile per avere una prima e grezza visualizzazione spaziale dell'insieme dei dati.

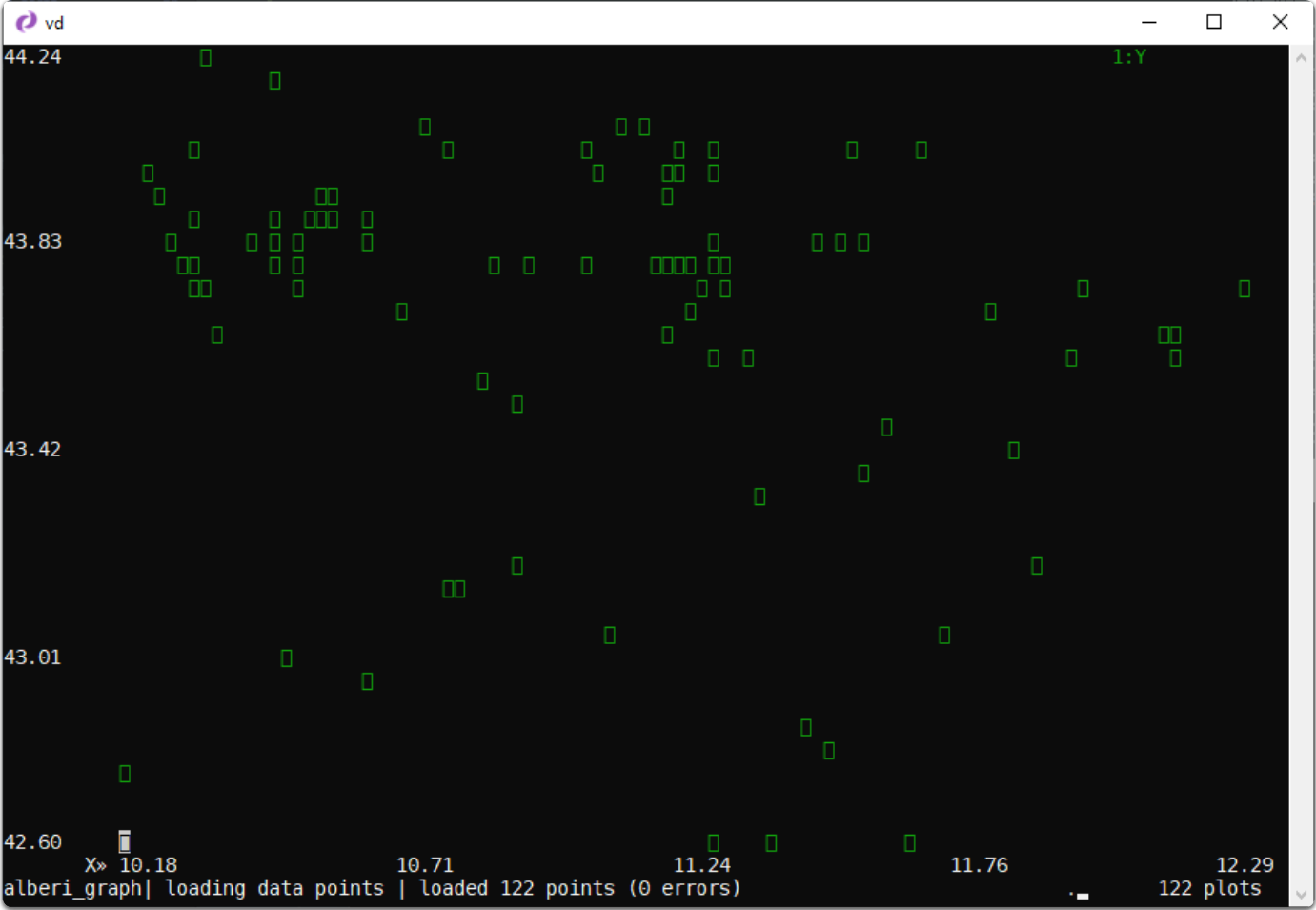
Il requisito per crearne uno, è avere due colonne numeriche, da usare come coppia di coordinate. Qualcosa come quella di sotto.

fid	X	Y
1	12.29	43.76
2	12.16	43.64
3	12.15	43.65
4	12.16	43.63
...	...	...

Per generare il grafico basterà seguire i seguenti passi:

- impostare come numeriche (numeri decimali digitando % sulla colonna, o # per numeri interi) le due colonne con le coordinate;
- impostare come colonna chiave la colonna con le ascisse, selezionandola e digitando ! ;
- selezionare la colonna con le ordinate e digitare . .

In output si avrà qualcosa come quella di sotto



## 13 Moduli Python

VisiData è un foglio elettronico basato su Python 3. Questo consente di utilizzare al suo interno la galassia dei moduli disponibili per Python.

Per farlo, bisogna precaricare questi moduli all'avvio di VisiData. Questo si realizza modificando il file di configurazione del programma - si chiama `.visidatarc` - contenuto nella cartella *home* dell'utente. Se si vuole ad esempio sfruttare il modulo `datetime`, si apre in modifica il file `.visidatarc` e si aggiunge la riga

```
from datetime import datetime
```

Poi si salva il file e si fa partire VisiData.

Se ad esempio si vuole fare un'analisi sulla data di immatricolazione per mese a partire dalla colonna `data_immatricolazione` e si volesse sfruttare il modulo `datetime`, si può procedere in questo modo:

- si va nella colonna `data_immatricolazione`;
- si preme `=`;
- si scrive nel prompt `datetime.strptime(data_immatricolazione, '%Y-%m-%d %H:%M:%S').month` e si pigia `Invio`.

Si ottiene qualcosa come quella di sotto:

alimentazione	kw	data immatricolazione	datetime.strptime(data immatrico...
BENZ	24	2005-12-21 00:00:00	12
BENZ	33	1995-04-03 00:00:00	4
B/OLIO	17	1986-01-01 00:00:00	1
BENZ	10	2001-11-09 00:00:00	11
BENZ	8	2000-11-10 00:00:00	11
BENZ	78	2000-07-19 00:00:00	7
BENZ	43	1999-08-11 00:00:00	8
BENZ	35	1987-05-06 00:00:00	5
BENZ	38	1979-01-01 00:00:00	1
BENZ	19	2015-07-03 00:00:00	7
GASOL	90	2000-02-23 00:00:00	2

E a questo punto si può rinominare la colonna premendo `^`, scrivendo "mese" e poi premendo `Invio`. E poi creare il foglio delle frequenze della colonna `mese`, pigiando `Shift+f` e scoprire che in Abruzzo si immatricola soprattutto a gennaio.

mese		count	#	percent	%	histogram
1	#	180903		15.46		*****
3	#	111084		9.49		*****
5	#	106995		9.14		*****
6	#	106055		9.06		*****
7	#	104239		8.91		*****
4	#	97355		8.32		*****
2	#	90660		7.75		*****
10	#	85932		7.34		*****
9	#	84515		7.22		*****
11	#	76751		6.56		*****
12	#	66368		5.67		*****
8	#	56549		4.83		*****

**NOTA BENE:** le funzioni del modulo `datetime` sono disponibili in modo nativo per le colonne già impostate (come tipo) a data. Quanto scritto sopra vale soprattutto come guida al come precaricare e utilizzare un modulo Python in VisiData.

# 14 Salvare un flusso di lavoro

## 14.0.0.1 Introduzione

In **VisiData** è possibile salvare il [log delle operazione fatte](https://www.visidata.org/docs/save-restore/) (<https://www.visidata.org/docs/save-restore/>) e riutilizzarlo per applicarle nuovamente al file di input.

Se ad esempio a partire da [questo file CSV](#) (sotto l'anteprima) si volessero cancellare tutte le righe che contengono il solo carattere `a` nella prima colonna e infine salvare l'*output*, la procedura è:

- `vd input.csv`;
- posizionarsi nella prima colonna e pigiare `|` per attivare la selezione tramite espressione regolare, scrivere `^a$` e premere `INVIO`. Verrà fatta la selezione descritta;
- pigiare in sequenza `g` e `d` per cancellare le righe selezionate;
- poi `CTRL+s`;
- e infine scegliere un nome di output (ad esempio `output.csv`) e pigiare `INVIO` per salvare il file.

field1	field2
a	0
b	3
a	4
c	5

Per salvare la procedura di sopra in un file di log, bisognerà premere `CTRL+d`, scegliere un nome per salvare il file (ad esempio `cancella_le_a.vd`) e pigiare `INVIO`.

Il file di log salvato avrà un contenuto come quello sottostante ([è un file TSV](#)), in cui sono "mappate" tutte le operazioni fatte:

sheet	col	row	longname	input	keystrokes	comment
global		null_value	set-option			
			open-file	input.csv	o	
input	field1		select-col-regex	^a\$		



sheet	col	row	longname	input	keystrokes	comment
input			delete-selected		gd	delete (cut) selected rows and move them to clipboard
input			save-sheet	output.csv	^S	save current sheet to filename in format determined by extension (default .tsv)

Per riapplicare la stessa procedura al file, senza interagire con lo schermo, il comando da lanciare è:

```
vd -b -p cancella_le_a.vd
```

- `-b` per eseguire VisiData senza attivarne l'interfaccia;
- `-p` per replicare un file di log.

Se questo comando si lancia dopo avere già creato il file di *output*, si avrà un messaggio di errore, che avvisa che il file esiste già e non può essere sovrascritto.

Per fare in modo che venga sovrascritto bisognerà aggiungere l'opzione `-y` :

```
vd -y -b -p cancella_le_a.vd
```

A schermo verrà restituito l'elenco delle operazione svolte:

```
opening cancella_le_a.vd as vd

```

14.0.0.2 Applicare lo stesso flusso a un file diverso, ma omologo

Spesso si ha la necessità di **applicare una stessa procedura ad altri file**, con la **stessa struttura**, ma con **contenuti differenti**.  
Per utilizzare il file di log creato con un file diverso da `input.csv`, bisognerà modificare il file di log (si modifica con qualsiasi editor di testo):

- rimuovere la riga con il riferimento a `input.csv`
- rimuovere il riferimento al nome del foglio (qui è `input`, come il nome del file).

Diventerà quindi:

sheet	col	row	longname	input	keystrokes	comment
global		null_value	set-option			
	field1		select-col-regex	^a\$		
			delete-selected		gd	delete (cut) selected rows and move them to clipboard

sheet	col	row	longname	input	keystrokes	comment
			save-sheet	output.csv	^S	save current sheet to filename in format determined by extension (default .tsv)

Fatto questo, il comando da usare con un nuovo file denominato ad esempio `input_nuovo.csv` sarà:

```
vd -y -b -p cancella_le_a.vd input_nuovo.csv
```

In output verrà creato il file `output.csv`.

Se si vuole creare un file di output con nome diverso, si deve cancellare dal file di log la riga in cui si fa riferimento a `output.csv` e lanciare il comando:

```
vd -y -b -p cancella_le_a.vd input_nuovo.csv -o output_nuovo.csv
```

#### 14.0.0.3 Nota sul file di log

Si tratta di un TSV, quindi è possibile leggerlo e modificarlo proprio con VisiData:

# 15 Soluzione problemi

## 15.0.0.1 Riga attiva di colore nero

Quando si usa VisiData su Windows Subsystem for Linux, la riga attiva risulta invisibile perché tutta nera.

Per fare in modo che torni "visibile" bisogna modificare il file di configurazione di VisiData. Questo file di *default* non esiste: si deve trovare nella home dell'utente (quindi in `~`) e si deve nominare come `.visidatarc`.

Questi i passi:

```
## vai nella cartella home dell'utente
cd ~
## nano è un editor di testo, si installa con "sudo apt-get install nano".
## Si può usare qualsiasi editor per modificare il file
nano .visidatarc
```

Si aprirà il file `.visidatarc` a cui bisognerà aggiungere le stringhe sottostanti e poi salvare il file.

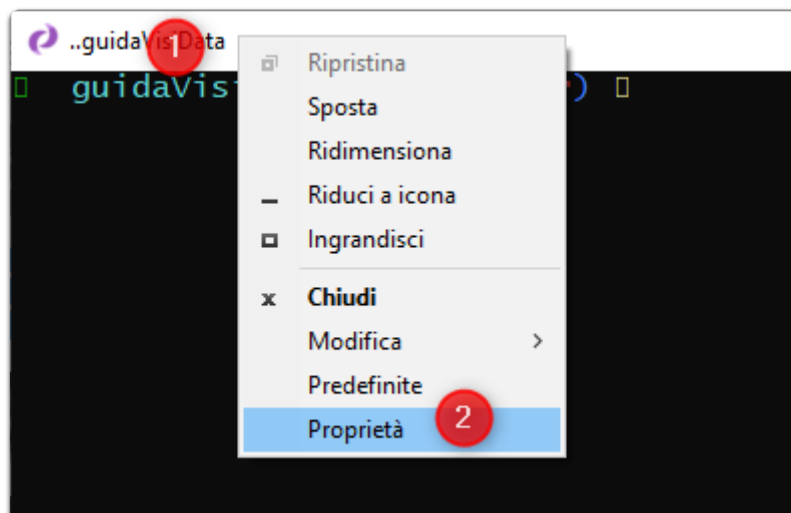
```
options.color_key_col=' '
options.color_selected_row='yellow'
options.color_note_type='yellow'
options.color_graph_hidden='blue'
options.color_column_sep='blue'
options.null_value=""
options.color_key_col="blue"
```

## 15.0.0.2 Caratteri non leggibili in Windows Subsystem for Linux

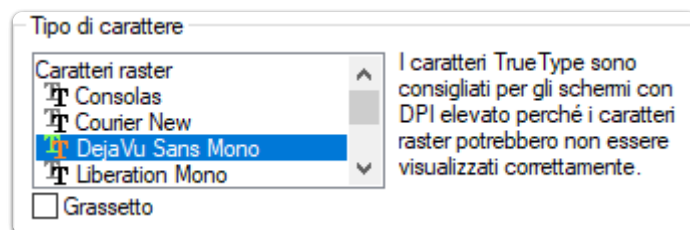
Se si usa VisiData in Windows Subsystem for Linux, alcuni caratteri possono risultare non leggibili. Questo dipende dal font usato.

Per cambiarlo:

1. click con il destro del mouse sulla barra;
2. e scegliere Proprietà.



Infine si consiglia il font di sotto:



# 16 Ricette

## 16.0.0.1 Salvare una tabella HTML in CSV, a partire da una pagina web

In questo esempio il presupposto è che si voglia trasformare in CSV una tabella HTML presente in una pagina web, in cui ci sono più tabelle. Il punto di partenza è quello di estrarre dalla pagina sorgente, una pagina HTML che contiene soltanto la tabella di interesse.

La pagina di esempio è:

[https://web.archive.org/web/20190618071304/http://www.sias.regione.sicilia.it/NHEOWLH140\\_00\\_1.html](https://web.archive.org/web/20190618071304/http://www.sias.regione.sicilia.it/NHEOWLH140_00_1.html)  
([https://web.archive.org/web/20190618071304/http://www.sias.regione.sicilia.it/NHEOWLH140\\_00\\_1.html](https://web.archive.org/web/20190618071304/http://www.sias.regione.sicilia.it/NHEOWLH140_00_1.html))

Questa è composta da diverse tabelle.



Quella di interesse è l'unica composta da più di 7 colonne, e la query XPATH per estrarla è

```
//table[count(tr/td)>7].
```

La pagina di origine ha l'encoding in ISO-8859-1, ed è da convertire in UTF-8.

Per scaricare la pagina, cambiarne l'encoding ed estrarre la tabella di interesse il comando può essere

```
curl "http://www.sias.regione.sicilia.it/NHEOWLH140_00_1.html" | \
iconv -f ISO-8859-1 -t utf-8 | \
scrape -be '//table[count(tr/td)>7]'
```

L'utilità usata sopra per fare la query XPATH è `scrape-cli` (<https://github.com/aborruso/scrape-cli>).

In output si ha una pagina web che contiene soltanto la tabella di interesse.

Per passarla a VisiData bisognerà modificare così lo script:

```
curl "http://www.sias.regione.sicilia.it/NHEOWLH140_00_1.html" | \
iconv -f ISO-8859-1 -t utf-8 | \
scrape -be '//table[count(tr/td)>7]' | \
vd -f html
```

Come risultato si aprirà la finestra di sotto. Per aprire la tabella sarà necessario premere INVIO e poi si potrà salvare la tabella in CSV (o altri formati) digitando CTRL+ S.

	id	nrows	#	ncols	#	classes
table	2	95		13		2

Se si vuole inserire nello script anche la procedura di salvataggio, bisogna modificare lo script in questo modo:

```
curl "http://www.sias.regione.sicilia.it/NHEOWLH140_00_1.html" | \
iconv -f ISO-8859-1 -t utf-8 | \
scrape -be '//table[count(tr/td)>7]' | \
vd -b -f html -o out.csv -p dive.vd
```

Note:

- `-b` è per eseguire VisiData senza interfaccia;
- `-p dive.vd` per eseguire dei comandi (qui semplicemente l'apertura della tabella a partire dalla finestra iniziale di sopra).

Nel file `dive.vd` c'è il seguente contenuto.

```
sheet  col row longname  input  keystrokes  comment
      open-file  -    o
-      0  dive-row      ^J
```

Il file di sopra è un file di log di VisiData (vedi [sezione dedicata](#)). Tutte le operazioni fatte in VisiData finiscono in un log che può essere visualizzato e salvato. Quindi è possibile salvare la visualizzazione dell'elenco delle tabelle e poi l'apertura di quella di interesse.

Per creare questo di sopra:

- arrivare a visualizzare la tabella su VisiData;
- digitare `INVIO`;
- digitare `SHIFT + d` per visualizzare il log dei comandi;
- e infine `CTRL + s` per salvare il file `dive.vd`.

## 17 Fare proposte e/o Chiedere supporto

Il luogo dove farlo è senz'altro questo <https://github.com/saulpw/visidata/issues> (<https://github.com/saulpw/visidata/issues>). È bene farlo scrivendo sempre la versione di VisiData utilizzata, allegare i dati per replicare eventuali problemi riscontrati, documentare la procedura che si vuole realizzare.



## 18 Note finali

Il grande **Jeremy Singer Vine** lo usa da molto più tempo di me e ha creato la migliore guida introduttiva dedicata in inglese <https://jsvine.github.io/intro-to-visidata/> (<https://jsvine.github.io/intro-to-visidata/>), di cui sono felice di essere nei [ringraziamenti](https://jsvine.github.io/intro-to-visidata/#acknowledgments) (<https://jsvine.github.io/intro-to-visidata/#acknowledgments>). Ne ho attinto a piene mani per scrivere questo post e colgo l'occasione per **ringraziarlo di cuore** pubblicamente.

È un'applicazione molto bella, che probabilmente per alcuni diverrà (come è adesso per me) di uso quotidiano. **Jeremy** (<https://jsvine.github.io/intro-to-visidata/the-big-picture/what-is-visidata/#why-not-use-visidata>) afferma che per analisi complesse, analisi geospaziali, *data literature* è meglio usare altro. Mi sembra sensato.

Non sostituisce R o Pandas, ma li può certamente affiancare e spesso precedere nel workflow di lavoro sui dati. È un altro strumento che consiglio di aggiungere nella cassetta degli attrezzi.

Lo sviluppatore principale e chi collabora con lui sono molto attenti alle richieste e alle proposte che vengono dall'esterno, in una bella mescolanza di sensibilità "tecnologica" e umana.

In queste settimane ho aperto diverse *issue* a cui è stato dato sempre riscontro. Alcune erano richieste da uomo primitivo, altre erano buone: confesso di essere **orgoglioso** di avere avuto degli effetti nel rilascio dell'ultima versione (la 1.5.1) e di essere citato più volte nelle [note di rilascio](https://github.com/saulpw/visidata/releases/tag/v1.5.1) (<https://github.com/saulpw/visidata/releases/tag/v1.5.1>). Ringrazio molto **Saul Pwanson** (<http://saul.pw/>) e **Anja Kefala** (<https://github.com/anjakefala>).

~~A breve gli proporrò l'integrazione in lettura e scrittura dei **datapackage** (<https://frictionlessdata.io/specs/data-package/>), in modo da estendere la qualità e le modalità di lavoro su dati ben descritti e strutturati. (fatto (<https://github.com/saulpw/visidata/issues/237>))~~

Qui vorremmo **aggiornare** e **arricchire** nel tempo i testi e si accettano anche **proposte** di **integrazioni**, **correzioni** e **modifiche**.

## 19 URL consigliati

- la guida di **Jeremy Singer Vine** <https://jsvine.github.io/intro-to-visidata/> (<https://jsvine.github.io/intro-to-visidata/>)
- le ricette sul sito ufficiale <http://visidata.org/docs/> (<http://visidata.org/docs/>) ;
- la documentazione ufficiale <http://visidata.org/man/> (<http://visidata.org/man/>) ;
- la playlist YouTube a cura dell'autore <https://www.youtube.com/playlist?list=PLxu7QdBkC7drrAGfYzatPGVHlpv4Et46W> (<https://www.youtube.com/playlist?list=PLxu7QdBkC7drrAGfYzatPGVHlpv4Et46W>) ;
- la lista visuale degli shortcut da tastiera <http://visidata.org/docs/kblayout/> (<http://visidata.org/docs/kblayout/>) .

## II. Esempi



**NOTA BENE:** Questa pagina si deve a [Salvatore Fiandaca](https://twitter.com/totoflandaca) (<https://twitter.com/totoflandaca>), che ha tradotto in italiano [10 Ways to Use VisiData](https://www.visidata.org/blog/2020/ten/) (<https://www.visidata.org/blog/2020/ten/>). Grazie.

## 20 10 Modi di usare Visidata

### 20.1 1. esplora i set di dati senza sforzo, indipendentemente dal formato

```
$ vd albums.json dropbox/*.xls* agents.sqlite
```

Utilizzare i tasti freccia (↑↓→←) per spostarsi e **q** per uscire.

**VisiData** carica i dati da [molte fonti e formati diversi](https://www.visidata.org/formats) (<https://www.visidata.org/formats>), incluso [tutto ciò che i pandas possono caricare!](https://pandas.pydata.org/pandas-docs/stable/reference/io.html) (<https://pandas.pydata.org/pandas-docs/stable/reference/io.html>)

I nuovi caricatori sono inclusi in ogni versione e puoi [crearne uno tu stesso facilmente](https://www.visidata.org/docs/loaders). (<https://www.visidata.org/docs/loaders>)

### 20.2 2. convertire da qualsiasi formato di input in qualsiasi formato di output

All'interno di **VisiData**, premere **Ctrl+S** per salvare il file nel formato specificato dall'estensione.

Converti direttamente dalla riga di comando con:

```
$ vd -b input.csv -o output.json
```

### 20.3 3. come sostituto interattivo di grep, awk, sed, cut, sort, uniq

```
$ netstat -an | vd -f fixed -o used-ports.txt
```

Premi **Ctrl+Q** per uscire da **VisiData** e salvare nel file di output specificato.

### 20.4 4. grattare i dati della tabella HTML da una pagina Web:

```
$ vd https://en.wikipedia.org/wiki/List_of_largest_cities
```

## 20.5 5. vedere una differenza visiva di due fogli con la stessa struttura

```
$ vd --diff foo1.csv foo2.csv
```

## 20.6 6. esplora un oggetto dal Python REPL

```
>>> visidata.view(obj)
```

## 20.7 7. esplora tutti i pandas DataFrame

```
>>> visidata.view_pandas(df)
```

## 20.8 8. creare una pipeline di dati ad hoc

**VisiData** può anche essere utilizzato in modalità batch con [script \(https://www.visidata.org/docs/save-restore/\)](https://www.visidata.org/docs/save-restore/) senza richiedere alcuna interazione.

Salvare il *cmdlog* in un file *.vd* con `Ctrl+D foo.vd`, quindi riprodurre il file *.vd* salvato:

```
$ vd -b -p foo.vd
```

## 20.9 9. rinominare un mucchio di file

Avviare *vd* nella directory con i file Modifica i nomi dei file ([documentazione sui comandi di modifica \(https://www.visidata.org/docs/edit/\)](https://www.visidata.org/docs/edit/)) `Ctrl+S` per eseguire il commit ([guarda questo video che mostra come utilizzare il browser di directory \(https://www.youtube.com/watch?v=I2Bpmm0yAGw\)](https://www.youtube.com/watch?v=I2Bpmm0yAGw))

## 20.10 10. vedi tutte le opzioni e i comandi disponibili

Esistono molti altri modi per utilizzare **VisiData**! Per saperne di più, premi `Ctrl H` per vedere la [manpage \(https://www.visidata.org/man\)](https://www.visidata.org/man) o `z Ctrl+H` per vedere l'elenco completo dei comandi disponibili sul foglio corrente.

[torna alla guida](#)

[scritto da Saul Pwanson 2020-05-01]

---

**VisiData** © 2017-2020 [Saul Pwanson \(https://www.saul.pw/\)](https://www.saul.pw/) | Licensed under [GPLv3 \(https://github.com/saulpw/visidata/blob/stable/LICENSE.gpl3\)](https://github.com/saulpw/visidata/blob/stable/LICENSE.gpl3) | [Privacy Policy \(https://www.visidata.org/privacy\)](https://www.visidata.org/privacy)

## 21 Come salvare un insieme di operazioni e applicarle come modello ad altri file

### 21.1 Introduzione

In **VisiData** è possibile salvare il [log delle operazioni fatte](https://www.visidata.org/docs/save-restore/) (<https://www.visidata.org/docs/save-restore/>) e riutilizzarlo per applicarle nuovamente al file di input.

Se ad esempio a partire da [questo file CSV](#) (sotto l'anteprima) si volessero cancellare tutte le righe che contengono il solo carattere `a` e infine salvare l'*output*, la procedura è:

- `vd input.csv`;
- posizionarsi nella prima colonna e pigiare `|` per attivare la selezione tramite espressione regolare, scrivere `^a$` e premere `INVIO`. Verrà fatta la selezione descritta;
- pigiare in sequenza `g` e `d` per cancellare le righe selezionate;
- poi `CTRL+s`;
- e infine scegliere un nome di output (ad esempio `output.csv`) e pigiare `INVIO` per salvare il file.

field1	field2
a	0
b	3
a	4
c	5

Per salvare la procedura di sopra in un file di log, bisognerà premere `CTRL+d`, scegliere un nome per salvare il file (ad esempio `cancella_le_a.vd`) e pigiare `INVIO`.

Il file di log salvato avrà un contenuto come quello sottostante ([è un file TSV](#)), in cui sono "mappate" tutte le operazioni fatte:

```
sheet  col row longname  input  keystrokes  comment
global      null_value set-option
          open-file  input.csv  o
input  field1      select-col-regex  ^a$ |  select rows matching regex in current column
input          delete-selected      gd  delete (cut) selected rows and move them to clipboard
input          save-sheet  output.csv  ^S  save current sheet to filename in format determined
by extension (default .tsv)
```

Per riapplicare la stessa procedura al file, senza interagire con lo schermo, il comando da lanciare è:

```
vd -b -p cancella_le_a.vd
```

- `-b` per eseguire VisiData senza attivarne l'interfaccia;
- `-p` per replicare un file di log.

Se questo comando si lancia dopo avere già creato il file di *output*, si avrà un messaggio di errore, che avvisa che il file esiste già e non può essere sovrascritto.

Per fare in modo che venga sovrascritto bisognerà aggiungere l'opzione `-y`:

```
vd -y -b -p cancella_le_a.vd
```

A schermo verrà restituito l'elenco delle operazioni svolte:

```
opening cancella_le_a.vd as vd
"input.csv"
opening input.csv as csv
select rows matching regex in current column
"^a$"
search wrapped
2 matches for /^a$/
selected 2 rows
delete (cut) selected rows and move them to clipboard
copied 2 rows to clipboard
deleted 2 rows
save current sheet to filename in format determined by extension (default .tsv)
"output.csv"
saving 1 sheets to output.csv as csv
replay complete
```

## 21.2 Applicare le stesse procedure a un file diverso, ma omologo

Spesso si ha la necessità di **applicare una stessa procedura ad altri file**, con la **stessa struttura**, ma con **contenuti differenti**.

Per utilizzare il file di log creato con un file diverso da `input.csv`, bisognerà modificare il file di log (si modifica con qualsiasi editor di testo):

- rimuovere la riga con il riferimento a `input.csv`
- rimuovere il riferimento al nome del foglio (qui è `input`, come il nome del file).

Diventerà quindi:

```
sheet  col row longname  input  keystrokes  comment
global      null_value set-option
      field1      select-col-regex  ^a$ |  select rows matching regex in current column
              delete-selected      gd  delete (cut) selected rows and move them to clipboard
              save-sheet  output.csv  ^S  save current sheet to filename in format determined by
extension (default .tsv)
```

Fatto questo, il comando da usare con un nuovo file denominato ad esempio `input_nuovo.csv` sarà:

```
vd -y -b -p cancella_le_a.vd input_nuovo.csv
```

In output verrà creato il file `output.csv`.

Se si vuole creare un file di output con nome diverso, si deve cancellare dal file di log la riga in cui si fa riferimento a `output.csv` e lanciare il comando:

```
vd -y -b -p cancella_le_a.vd input_nuovo.csv -o output_nuovo.csv
```

## 21.3 Nota sul file di log

Si tratta di un TSV, quindi è possibile leggerlo con un client dedicato. Quello di sopra ad esempio è:

sheet	col	row	longname	input	keystrokes	comment
global		null_value	set-option			
	field1		select-col-regex	^a\$		
			delete-selected		gd	delete (cut) selected rows and move them to clipboard
			save-sheet	output.csv	^S	save current sheet to filename in format determined by extension (default .tsv)



## 22 Informativa privacy estesa

In linea con la legislazione Europea, è fondamentale che ogni utente del sito web comprenda cosa sono i cookie e per quale motivo vengono utilizzati, in modo che gli utenti stessi possano decidere consapevolmente se accettarne l'utilizzo o no sui loro dispositivi.

**Un cookie è un piccolo file di testo contenente un numero di identificazione univoco che viene trasferito dal sito web sul disco rigido del vostro computer attraverso un codice anonimo in grado di identificare il computer ma non l'utente e di monitorare passivamente le vostre attività sul sito.**

I cookie possono essere classificati in due categorie, in base alle loro funzionalità:

- Cookie tecnici (di cui fanno parte i cookie sulle prestazioni e i cookie funzionali)
- Cookie profilanti (di cui fanno parte i cookie legati alle pubblicità)

Il sito utilizza solo il primo tipo di cookie, di tipo tecnico, per i quali non è richiesto alcun consenso.

Nel dettaglio vengono utilizzati le seguenti tipologie di cookie:

### **Cookie strettamente necessari**

Questi cookie sono essenziali al fine di permettere la navigazione del sito web e l'utilizzo delle sue funzionalità, come ad esempio l'accesso ad alcune aree protette. Senza questi cookie, alcune funzionalità richieste come ad esempio il login al sito o l'inserimento degli articoli da parte dei collaboratori non potrebbero essere fornite.

### **Cookie sulle prestazioni**

Questi cookie raccolgono informazioni su come gli utenti utilizzano il sito web, ad esempio quali pagine vengono visitate più spesso e se gli utenti ricevono messaggi di errore da queste pagine. Questi cookie non raccolgono informazioni che identificano un visitatore specifico. Tutte le informazioni raccolte da questi cookie sono aggregate e quindi anonime. Vengono utilizzate unicamente per migliorare il funzionamento del sito web.

### **Cookie di analisi di traffico anonimizzati**

Questo sito web utilizza inoltre i **cookie di Google Analytics**, un servizio di analisi web fornito da Google. Le informazioni generate dal cookie sull'utilizzo del sito web da parte vostra (compreso il vostro indirizzo IP) verranno trasmesse e depositate presso i server di Google che utilizzerà queste informazioni allo scopo di tracciare ed esaminare il vostro utilizzo del sito web, compilare report sulle attività svolte sul sito web e fornire altri servizi relativi alle attività e all'utilizzo di Internet. I dati generati da Google Analytics sono conservati da Google così come indicato nella Informativa reperibile al seguente link.

L'utente può disabilitare in modo selettivo l'azione di Google Analytics installando sul proprio browser la componente di opt-out fornito da Google. Per disabilitare l'azione di Google Analytics, si rinvia al link di seguito indicato:

<https://tools.google.com/dlpage/gaoptout>

## Cookie funzionali

Questi cookie consentono al sito web di ricordare le scelte che avete effettuato (come il vostro username, la vostra lingua o l'area geografica in cui vivete) al fine di ottimizzare e fornire funzionalità più avanzate. Questi cookie possono inoltre essere utilizzati per fornire funzionalità da voi richieste come ad esempio la visualizzazione di un video o la possibilità di commentare sul sito. Queste informazioni raccolte dai cookie possono essere anonime e non devono tracciare la navigazione e le attività dell'utente su altri siti web. Vi preghiamo di considerare che alcune funzionalità del sito web potrebbero essere ridotte o non disponibili disabilitando i cookie. Nel dettaglio, gli utenti possono controllare, eliminare o impedire il caricamento di cookie agendo sui loro browser così come sotto indicato.

## 22.1 Cookie di terze parti

### Google Analytics

Il Sito impiega Google Analytics. Si tratta di un servizio di analisi web fornito da Google Inc. ("Google") che utilizza cookie depositati sul computer dell'utente per consentire analisi statistiche in forma aggregata in ordine all'utilizzo del sito web visitato.

I dati generati da Google Analytics sono conservati da Google così come indicato nella Informativa reperibile al seguente link – <https://developers.google.com/analytics/devguides/collection/analyticsjs/cookie-usage> Per consultare l'informativa privacy della società Google Inc., titolare autonomo del trattamento dei dati relativi al servizio Google Analytics, si rinvia al sito Internet.

### Google Fonts (What does using the Google Fonts API mean for the privacy of my users?)

([https://developers.google.com/fonts/faq#what\\_does\\_using\\_the\\_google\\_fonts\\_api\\_mean\\_for\\_the\\_privacy\\_of\\_my\\_users](https://developers.google.com/fonts/faq#what_does_using_the_google_fonts_api_mean_for_the_privacy_of_my_users))

Questo sito web utilizza API di Google Fonts, l'uso di Google Fonts non è autenticato. Nessun cookie viene inviato dai visitatori del sito web all'API di Google Fonts. Le richieste all'API di Google Fonts vengono eseguite su domini specifici delle risorse, come [fonts.googleapis.com](https://fonts.googleapis.com) o [fonts.gstatic.com](https://fonts.gstatic.com), in modo che le richieste per i caratteri siano separate e non contengano credenziali inviate a google.com mentre utilizzando altri servizi Google autenticati, come Gmail. Le richieste per le risorse CSS vengono memorizzate nella cache per 1 giorno. Per ulteriori informazioni sulle informazioni raccolte da Google e su come vengono utilizzate e protette, consulta le [Norme sulla privacy di Google](https://policies.google.com/privacy) (<https://policies.google.com/privacy>)

## 22.2 Widget Social Network

I widget dei social network sono quei particolari "pulsanti" o "blocchi" presenti sul sito che raffigurano le icone di social network (esempio, Facebook e Twitter) e consentono agli utenti che stanno navigando di interagire con un "click" direttamente con le piattaforme social. Si riportano i link ove l'utente può prendere visione dell'informativa privacy relativa alla gestione dei dati da parte dei social cui i pulsanti e le widget rinviano. Nessuna informazione viene condivisa dal sito in cui il widget è incorporato.

Per maggiori informazioni, anche sulla disattivazione di tali cookie, si consiglia di consultare i seguenti link:

- [Twitter \(http://twitter.com/privacy\)](http://twitter.com/privacy)
- [Facebook \(https://www.facebook.com/help/cookies\)](https://www.facebook.com/help/cookies)
- [LinkedIn \(https://www.linkedin.com/legal/cookie\\_policy\)](https://www.linkedin.com/legal/cookie_policy)
- [Google Plus \(http://www.google.it/intl/it/policies/technologies/cookies/\)](http://www.google.it/intl/it/policies/technologies/cookies/)
- [AddThis \(http://www.addthis.com/privacy/privacy-policy\)](http://www.addthis.com/privacy/privacy-policy)

## 22.3 Pubblicità

Sono presenti annunci di pubblicità etica di **Read the Docs**

(<https://docs.readthedocs.io/en/latest/advertising/index.html>) che rispetta la privacy degli utenti.

La pubblicità etica è la fonte principale del progetto **Read the Docs** (<https://docs.readthedocs.io/en/latest/index.html>)

## 22.4 Come disabilitare i cookie mediante configurazione del browser

Fermo restando quanto sopra indicato in ordine ai cookie strettamente necessari alla navigazione, l'utente può eliminare gli altri cookie attraverso la funzionalità a tal fine messa a disposizione dal Titolare tramite la presente informativa oppure direttamente tramite il proprio browser. Ciascun browser presenta procedure diverse per la gestione delle impostazioni. L'utente può ottenere istruzioni specifiche attraverso i link sottostanti.

- [Google Chrome \(https://support.google.com/accounts/answer/61416?hl=en\)](https://support.google.com/accounts/answer/61416?hl=en)
- [Mozilla Firefox \(https://support.mozilla.org/en-US/kb/enable-and-disable-cookies-website-preferences\)](https://support.mozilla.org/en-US/kb/enable-and-disable-cookies-website-preferences)
- [Internet Explorer \(http://windows.microsoft.com/it-it/windows-vista/block-or-allow-cookies\)](http://windows.microsoft.com/it-it/windows-vista/block-or-allow-cookies)
- [Safari \(http://support.apple.com/kb/PH17191?viewlocale=it\\_IT&locale=it\\_IT\)](http://support.apple.com/kb/PH17191?viewlocale=it_IT&locale=it_IT)
- [Safari IOs \(http://support.apple.com/kb/HT1677?utm\\_source=Agillic%20Dialogue\)](http://support.apple.com/kb/HT1677?utm_source=Agillic%20Dialogue)
- [Opera \(http://help.opera.com/Windows/10.00/it/cookies.html\)](http://help.opera.com/Windows/10.00/it/cookies.html)

### Google Chrome

- Eseguire il Browser Chrome
- Fare click sul menù presente nella barra degli strumenti del browser a fianco della finestra di inserimento url per la navigazione
- Selezionare Impostazioni
- Fare clic su Mostra Impostazioni Avanzate
- Nella sezione "Privacy" fare clic su bottone "Impostazioni contenute"
- Nella sezione "Cookie" è possibile modificare le seguenti impostazioni relative ai cookie:
- Consentire il salvataggio dei dati in locale
- Modificare i dati locali solo fino alla chiusura del browser
- Impedire ai siti di impostare i cookie
- Bloccare i cookie di terze parti e i dati dei siti

- Gestire le eccezioni per alcuni siti internet
- Eliminazione di uno o tutti i cookie

**Mozilla Firefox** - Eseguire il Browser Mozilla Firefox - Fare click sul menù presente nella barra degli strumenti del browser a fianco della finestra di inserimento url per la navigazione - Selezionare Opzioni - Seleziona il pannello Privacy - Fare clic su Mostra Impostazioni Avanzate - Nella sezione "Privacy" fare clic su bottone "Impostazioni contenuti" - Nella sezione "Tracciamento" è possibile modificare le seguenti impostazioni relative ai cookie: - Richiedi ai siti di non effettuare alcun tracciamento - Comunica ai siti la disponibilità ad essere tracciato - Non comunicare alcuna preferenza relativa al tracciamento dei dati personali - Dalla sezione "Cronologia" è possibile: - Abilitando "Utilizza impostazioni personalizzate" selezionare di accettare i cookie di terze parti (sempre, dai siti più visitato o mai) e di conservarli per un periodo determinato (fino alla loro scadenza, alla chiusura di Firefox o di chiedere ogni volta) - Rimuovere i singoli cookie immagazzinati

### Internet Explorer

Eseguire il Browser Internet Explorer

- Fare click sul pulsante Strumenti e scegliere Opzioni Internet
- Fare click sulla scheda Privacy e nella sezione Impostazioni modificare il dispositivo di scorrimento in funzione dell'azione desiderata per i cookie:
- Bloccare tutti i cookie
- Consentire tutti i cookie
- Selezione dei siti da cui ottenere cookie: spostare il cursore in una posizione intermedia in modo da non bloccare o consentire tutti i cookie, premere quindi su Siti, nella casella Indirizzo Sito Web inserire un sito internet e quindi premere su Blocca o Consenti

### Safari

- Eseguire il Browser Safari
- Fare click su Safari, selezionare Preferenze e premere su Privacy
- Nella sezione Blocca Cookie specificare come Safari deve accettare i cookie dai siti internet.
- Per visionare quali siti hanno immagazzinato i cookie cliccare su Dettagli

### Safari IOs (Dispositivi Mobile)

- Eseguire il Browser Safari iOS
- Tocca su Impostazioni e poi Safari
- Tocca su Blocca Cookie e scegli tra le varie opzioni: "Mai", "Di terze parti e inserzionisti" o "Sempre"
- Per cancellare tutti i cookie immagazzinati da Safari, tocca su Impostazioni, poi su Safari e infine su Cancella Cookie e dati

### Opera

- Eseguire il Browser Opera
- Fare click sul Preferenze poi su Avanzate e infine su Cookie

- Selezionare una delle seguenti opzioni:
- Accetta tutti i cookie
- Accetta i cookie solo dal sito che si visita: i cookie di terze parti e che vengono inviati da un dominio diverso da quello che si sta visitando verranno rifiutati
- Non accettare mai i cookie: tutti i cookie non verranno mai salvati

Come indicato, l'Utente potrà negare il consenso all'installazione dei cookie per il mezzo del browser di navigazione in uso seguendo le istruzioni ivi fornite o accedendo al sito <http://www.youronlinechoices.com/it/le-tue-scelte/>.

Per ulteriori informazioni su come cancellare e controllare i cookie archiviati sul computer, si prega di consultare <http://www.allaboutcookies.org/manage-cookies/index>. Non possiamo tuttavia garantire la correttezza e l'aggiornamento delle informazioni ivi contenute.