TECNICHE E STRUMENTI PER LA VISUALIZZAZIONE DI DATI SCIENTIFICI

Giulio Botturi
Davide Quaglia

Contenuto

1. Visualizzazione grafica di serie di dati

Spesso è necessario visualizzare graficamente delle serie di dati, memorizzate e organizzate in basi di dati o file testuali, per una più chiara e immediata interpretazione.

1.1. Tecniche e strumenti

A questo scopo introduciamo ora alcune tecniche e strumenti utili ad automatizzare questa fase di analisi dei dati, ciascuno con le sue potenzialità, vantaggi e svantaggi. La scelta della tecnica dipende dal risultato che si vuole ottenere e dal grado di integrazione che si vuole avere a livello applicativo.

Principalmente i dati grezzi vengono rappresentati come tradizionali grafici per punti sul piano cartesiano, istogrammi, diagrammi circolari, a bolle e molti altri in parte derivati dai precedenti.

Gli strumenti illustrati in questa esercitazione sono tre: il software Gnuplot, la libreria Java JFreeChart e le primitive grafiche fornite da Java stesso.

In particolare Gnuplot non consente la realizzazione diretta di diagrammi circolari (grafico a torta). Per fare ciò è necessario realizzare complessi script che disegnano tutta la struttura della torta tramite funzioni matematiche oppure ricorrere, come segnalato nelle FAQ online sul sito del progetto Gnuplot, a una libreria C che sfrutta le GNU plotutils.

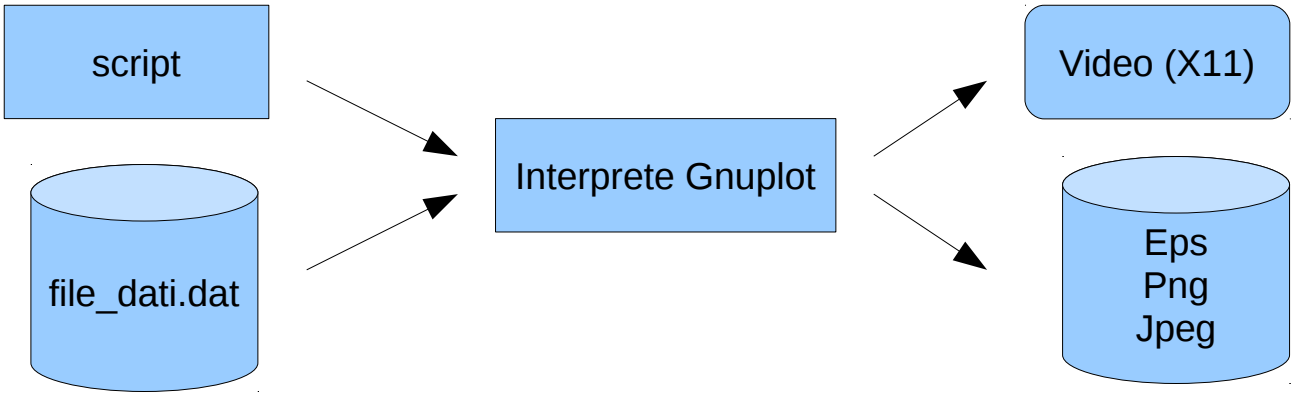
Vedremo nei due capitoli successivi, e in particolare in quello relativo alle primitive grafiche Java, come realizzare manualmente grafici in un linguaggio di programmazione. Questo ci consentirà di avere la massima libertà di personalizzazione, pagando il prezzo di dover gestire a livello geometrico e implementativo molti dettagli che strumenti automatici quali Gnuplot nascondono.

2. Gnuplot

Gnuplot è un versatile programma, distribuito gratuitamente in rete, per la realizzazione di grafici di funzioni matematiche e la rappresentazione grafica di dati grezzi.

Ai fini dell'esercitazione ci limitiamo ad analizzare la gestione dei dati grezzi memorizzati in semplici file di testo.

Lo schema di funzionamento di Gnuplot è il seguente:



L'interprete esegue i comandi di uno script che elabora i dati presenti in un file di testo. L'output viene mostrato a video oppure può essere dirottato su file nei più comuni formati di immagini.

Per avviare l'interprete, posizioniamoci nella directory in cui teniamo i file dati e script degli esempi e digitiamo il comando `gnuplot`.
I comandi possono essere direttamente impartiti all'interprete attraverso un prompt oppure essere letti da un file di script (ad esempio filescript.gp) con il comando

```
load “filescript.gp”
```

oppure è possibile direttamente passare il nome dello script in linea di comando alla chiamata di `gnuplot`

```
gnuplot filescript.gp
```

Per uscire dall'interprete si scrive il comando `quit`.

Nei seguenti paragrafi vediamo alcuni esempi di file di dati con i relativi script gnuplot per la loro visualizzazione. Tali sorgenti di dati verranno poi riutilizzati nei due capitoli successivi, per essere elaborati con altre tecniche e strumenti.

2.1. Coppie XY sul piano

Vediamo per prima cosa un semplice esempio in cui si vuole rendere graficamente l'andamento di due serie di dati affini, quali le temperature massime e minime giornaliere del mese di gennaio.
Tali informazioni sono memorizzate nel file *esempio1.dat* nel seguente formato:

```
# Temperature del mese di gennaio 2010
# giorno min max
01 -2 7
02 -2 6
# commento
03 0 7
04 -1 6
05 1 8
...
...
```

I valori sono memorizzati riga per riga, separati da uno o più spazi (o caratteri di tabulazione).

Abbiamo dunque bisogno di rappresentare su un piano in due dimensioni due serie di coppie di valori, (giorni – temperatura minima) e (giorno – temperatura massima).

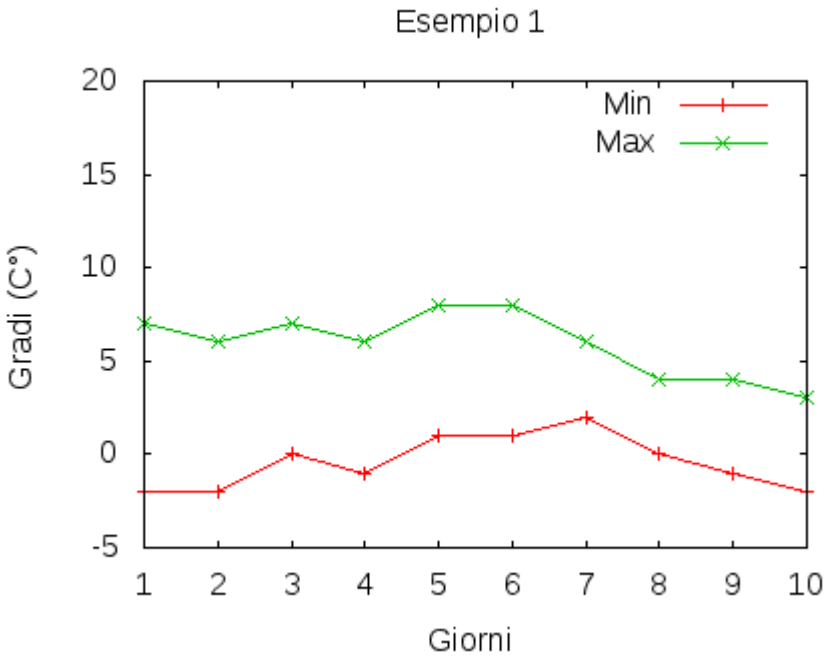
Analizziamo riga per riga lo script utilizzato per visualizzarli (*script1.gp)*

```
# questo è un commento
set title "Esempio 1"
set xlabel 'Giorni'
set ylabel 'Gradi (C°)'
set yrange [-5:20]
set style data linespoints
plot 'esempio1.dat' using 1:2 title 'Min', 'esempio1.dat' using 1:3 title 'Max'
pause -1
```

La prima riga indica il titolo da mostrare in cima al grafico.
La seconda e terza riga impostano le etichette da dare rispettivamente all'asse delle ascisse e delle ordinate.
La quarta riga imposta manualmente il range dei valori dell'asse delle ordinate. In mancanza di tale riga Gnuplot imposta automaticamente tale range a partire dal valore minimo visualizzato fino al massimo riscontrato. Nel nostro caso ampliare tale range ci consente una migliore resa grafica delle due spezzate che Gnuplot andrà a disegnare.

Le ultime due righe contengono i comandi più importanti.
In particolare il comando `set style data […]` permette di definire il tipo di grafico che si vuole ottenere. Molti sono gli stili possibili, qui ci limiteremo a vedere `linespoints e histogram`.
Il comando `plot` infine legge i dati grezzi dal file `esempio1.dat`, con la direttiva `using` indichiamo di utilizzare le colonne di dati 1 e 2, mentre con `title` diamo il nome `Min` alla curva che verrà mostrata sul grafico.
In questo esempio ricordiamo che sono due le serie di dati da mostrare insieme sullo stesso grafico, quindi dopo la virgola seguono le direttive per leggere i dati, questa volta dalle colonne 1 e 3, assegnando il nome `Max` alla curva generata.

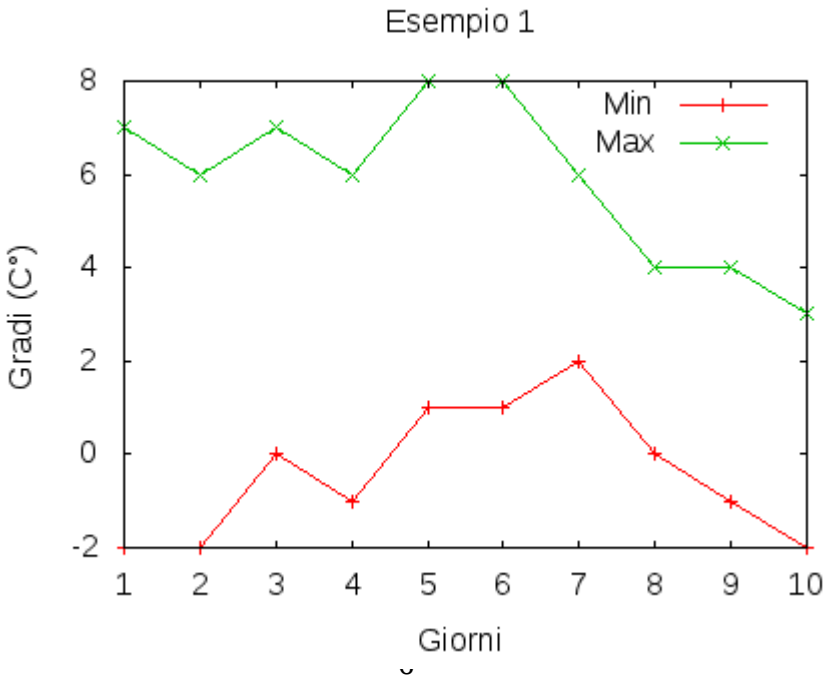
Di seguito il grafico che gnuplot genera dall'esecuzione dello script



Si noti il range ampliato sull'asse Y e la legenda in alto a destra.

Senza il comando `set yrange [-5:20]` il grafico risulterebbe così

```
# questo è un commento
set title "Esempio 1"
set xlabel 'Giorni'
set ylabel 'Gradi (C°)'
#set yrange [-5:20]
set style data linespoints
plot 'esempio1.dat' using 1:2 title 'Min',
     'esempio1.dat' using 1:3 title 'Max'
pause -1
```



2.2. Istogramma

Come secondo esempio vediamo la realizzazione di un istogramma a partire dal file di dati *esempio2.dat* strutturato nel seguento modo:

```
# Mese Entrate Uscite
Gennaio 4500 4350
Febbraio 4325 4100
Marzo 5200 4450
Aprile 4900 4600
Maggio 4620 4580
Giugno 4150 4450
```

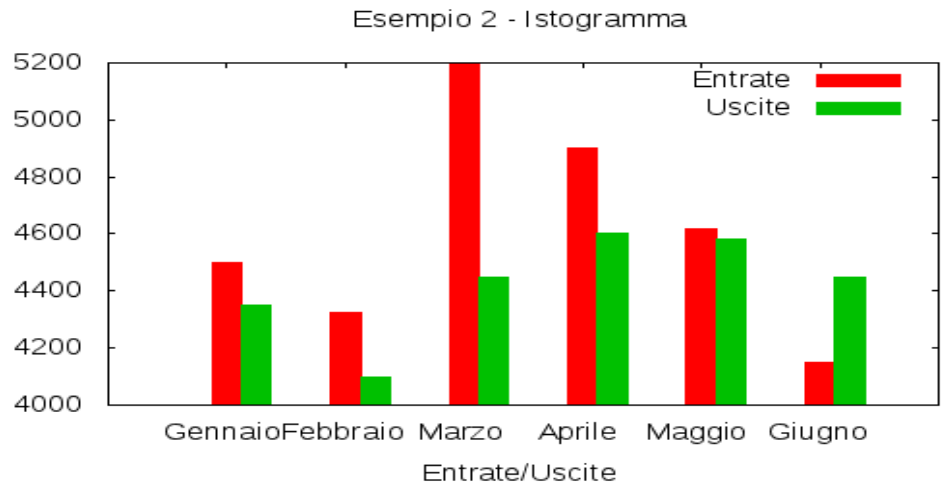
Sono memorizzate delle ipotetiche entrate e uscite (in euro) relative ai diversi mesi dell'anno e le vogliamo mettere a confronto graficamente. Anche in questo caso entrate e uscite rappresentano due serie di dati distinte, (mese – entrate) e (mese – uscite).

Analizziamo riga per riga lo script utilizzato (*script2.gp)*

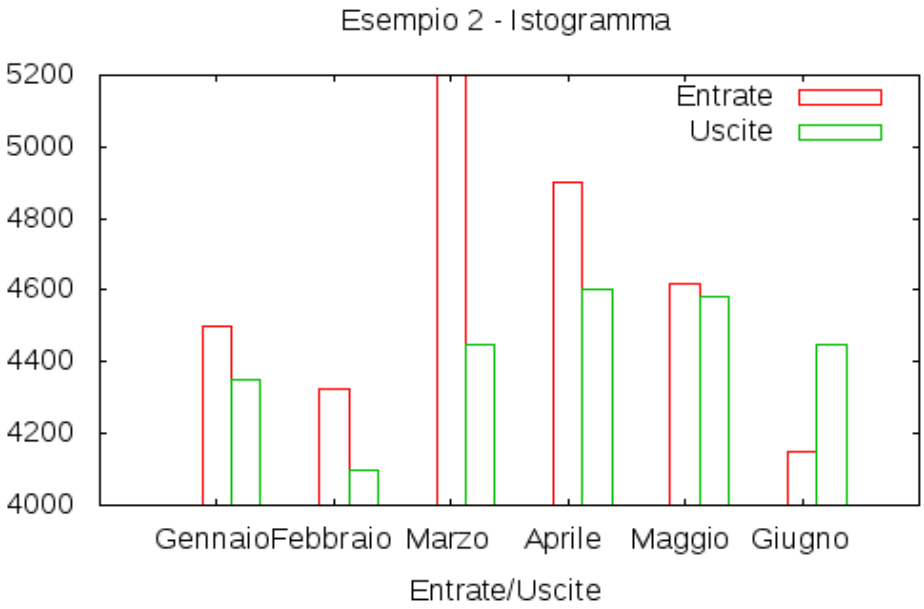
```
set title "Esempio 2 - Istogramma"
set xlabel 'Entrate/Uscite'
set style data histogram
set style fill solid
plot 'esempio2.dat' using 2:xticlabels(1) title 'Entrate', 'esempio2.dat' using
3:xticlabels(1) title 'Uscite'
pause -1
```

Come per l'esempio precedente, con la prima e seconda riga definiamo il titolo del grafico e l'etichetta da assegnare all'asse delle ascisse.
Il comando set style data histogram della terza riga imposta lo stile istogramma, specificando nella quarta riga la resa grafica da dare alle colonne (colorate al loro interno e non soltanto riquadrate come di default).
Il comando plot nuovamente legge due serie di dati dal file di testo, la prima dalle colonne 2 e 1, la seconda dalla colonne 3 e 1, rispettivamente con il nome 'Entrate' e 'Uscite'.
In questo caso i valori presente sull'asse X sono i mesi dell'anno, quindi vogliamo che sia indicato il nome testuale. Per questo motivo è necessario specializzare la direttiva `using` istruendola a piazzare i dati della colonna 1 come etichette di testo sotto l'asse X.

Di seguito il grafico che gnuplot genera dall'esecuzione dello script.



Senza il comando *`set style fill solid`* le colonne apparirebbero con la resa grafica di default



2.3. Gestire l'output degli script

L'output di default per gli script gnuplot è, in ambiente Linux, il terminale grafico X11 cioè una finestra a video.
E' possibile in alternativa redirigere tale output su file al fine di esportare i grafici sotto forma di immagini in vari formati per essere successivamente riutilizzati in altri ambiti. Per fare questo occorre inserire all'inizio dello script:

```
set terminal png size 450, 350
set output "esempio1bis.png"
```

Il primo comando indica a gnuplot che il terminale di output non è più X11 (valore di default), ma che deve provvedere a generare un file immagine in formato png (keyword alternative sono `jpeg gif eps`) di dimensione 450x350 pixel. Nella seconda riga indichiamo invece il nome del file di output in cui salvare tale immagine.
Di seguito è possibile richiamare gli altri comandi visti prima.

2.4. Esercizi

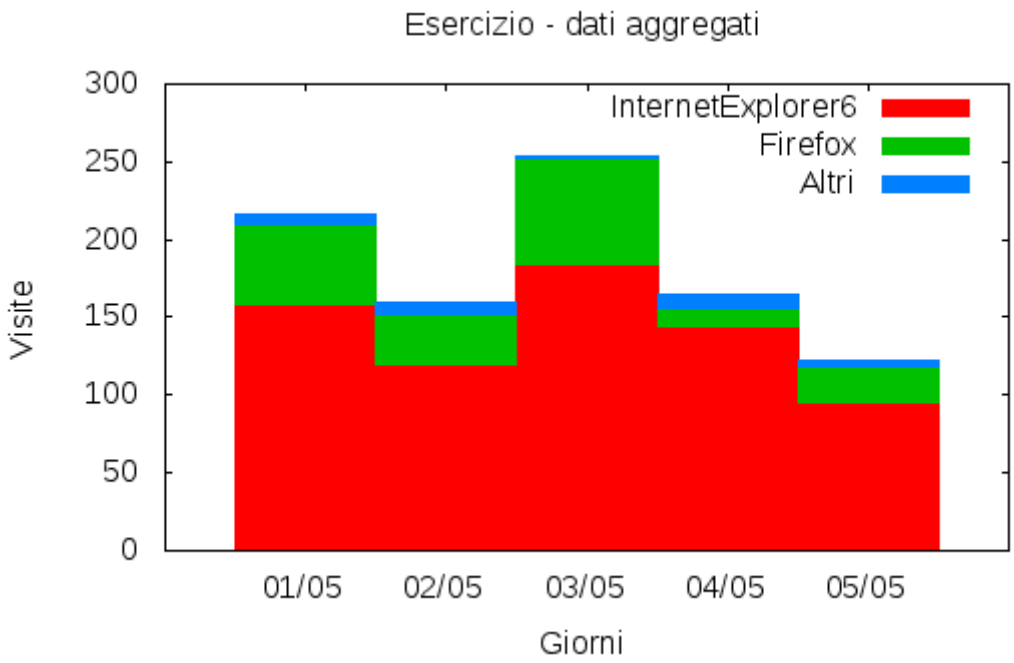
Per esercizio si chiede di realizzare un particolare tipo di istogramma utile per visualizzare dati aggregati.

Il file dati di esempio è il seguente (*esercizio.dat*):

```
# giorno IE6 firefox altri
01/05 158 52 6
02/05 120 32 8
03/05 184 68 2
04/05 144 12 9
05/05 95 23 4
```

in cui è memorizzato il numero di accessi giornaliero, ad un ipotetico sito web, suddiviso per la tipologia di browser.
Aggiungere la direttiva `set style histogram rowstacked` che costruisce delle barre verticali che aggregano i dati dei tre valori relativi al numero di visite. L'altezza della barra raggiunge la somma dei tre valori.
Aggiungere al grafico le etichette col nome dei due assi, “Giorni” per l'asse X e “Visite” per l'asse Y, oltre alla legenda relativa ai tre valori aggregati. Trattere i valori sull'asse X come stringhe testuali.

Il grafico che si dovrebbe ottenere è il seguente:



Per ottenere un aiuto su un comando è sufficiente scrivere sulla prompt dei comandi di Gnuplot

```
help nomecomando
```

Per ulteriori dettagli riguardo lo style histogram si rimanda al paragrafo 34 della documentazione ufficiale gnuplot, reperibile in formato pdf al link *http://www.gnuplot.info/documentation.html*.

3. Elaborazione elementare di immagini con Gimp

Nell'ambito dell'acquisizione e memorizzazione di dati scientifici e di laboratorio può essere necessario manipolare file di immagini per adattarle alle proprie esigenze o modificarne le caratteristiche fondamentali.
Quando trattiamo una immagine memorizzata con un certo formato, ci troviamo davanti ad un segnale digitale ottenuto da un segnale analogico in due dimensioni spaziali.
Tale trasformazione passa per le fasi di campionamento e quantizzazione che influenzano la qualità dell'immagine che si ottiene.

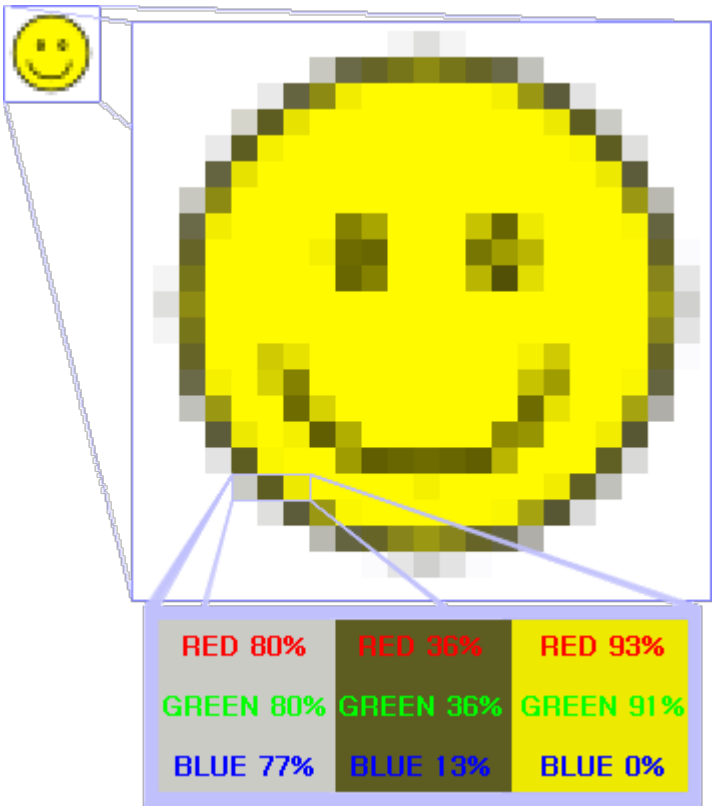
Una immagine digitale è quindi caratterizzata da una dimensione, ovvero il numero di pixel totale che la compongono, e dalla profondità di colore, cioè il numero di bit che rappresentano il colore di ciascun punto.
La dimensione in pixel è determinata dalla fase di campionamento mentre la profondità di colore dalla successiva fase di quantizzazione.

Un'altro aspetto da tenere in considerazione quando si trattano immagini, acquisite da dispositivi quali scanner o che si devono successivamente stampare, è la risoluzione, che collega il numero di pixel dell'immagine con le sue dimensioni di visualizzazione (su carta, su monitor).
Tale parametro è spesso indicato con l'unità di misura dpi (dot per inch) o ppi (point per inch) dove inch significa “pollice” e vale 2.54 centimetri.
Una risoluzione, ad esempio, di 300 dpi indica che il dispositivo è in grado di rilevare 300 campioni dall'immagine analogica per pollice. Stesso discorso vale per le stampanti, dove 300 dpi indicano che è in grado di stampare fedelmente in un pollice 300 pixel dell'immagine digitale.
Con risoluzione si intende dunque il numero di pixel contenuti nell'unità di misura considerata.

3.1. Gimp

Gimp (GNU Image Manipulation Program) è un programma di fotoritocco che permette di creare e modificare immagini raster. Immagine raster, termine che si significa trama/reticolo, si contrappone a immagine vettoriale, dove l'immagine stessa è definita con la descrizione degli elementi geometrici elementari che la compongono.

Questo è un esempio che mostra come invece è rappresentata una immagine raster.



Una volta aperto, il programma Gimp presenta la sua interfaccia con una finestra principale dove verrà caricata l'immagine da elaborare, e con due finestre secondarie contenti gli strumenti di disegno e altri box per la gestione dei colori, dei livelli e delle proprietà degli strumenti stessi.

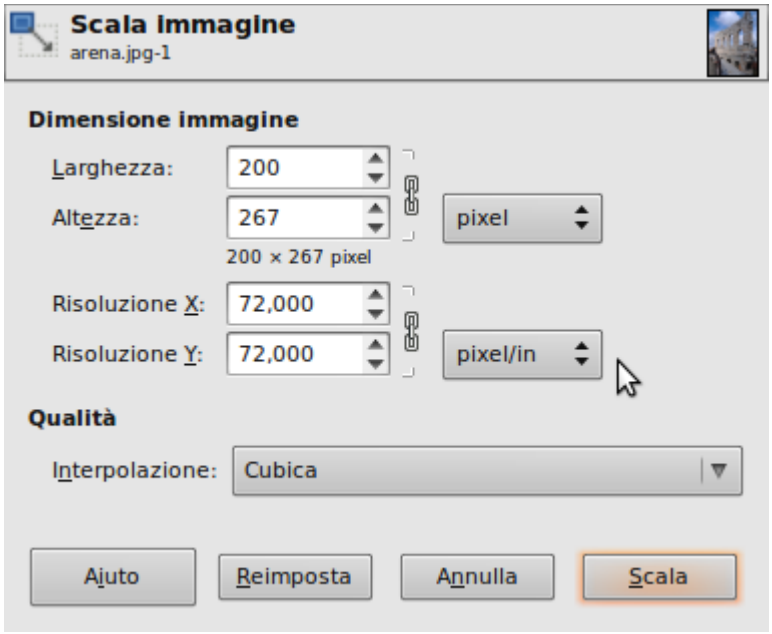
Una volta caricata una immagine la finestra ci mostra sulla barra del titolo, a fianco del nome, la sua dimensione in pixel.

Per vedere la risoluzione si può andare sul menu Immagine/Proprietà.

Nell'esempio l'immagine ha una dimensione di 768x1024 e una risoluzione di 72 ppi.

3.2. Scalare le dimensioni

Vogliamo ora ridurre la dimensione dell'immagine ad una larghezza di 200 pixel. Dal menu Immagine selezioniamo la voce “Scala immagine”.



Indicando nel campo “Larghezza” i 200 pixel desiderati, l'altezza viene aggiornata mantenendo la proporzione precedente tra le due misure. Applichiamo la modifica premendo “Scala”. Si noti che non viene alterati il contenuto della casella “Risoluzione” e quindi essa non cambia; il risultato stampato occuperà quindi un'area più piccola.

Si provi a caricare l'immagine dell'Arena di Verona, si riduca la larghezza da 768 a 200 pixel e si salvi il risultato in un altro file. Si provi a caricare il secondo file e a operare ora un aumento di dimensione, riportando l'immagine ai 768 pixel di larghezza come in origine; nel campo “Interpolazione” si selezioni “Nessuna”. Si confronti il risultato con l'immagine del file originale e si valuti le differenze. Da cosa son causati i difetti riscontrati ?

Attenzione! L'operazione di ridimensionamento di una immagine è irreversibile, in quanto nel momento in cui la si riduce si perde informazione che non si può più recuperare riportando l'immagine alla dimensione originale.

I difetti sono causati dalla perdita di informazione dovuta alla riduzione della dimensione da 768 a 200. Gimp prova a sopperire alla mancanza di informazione interpolando i dati sul colore dei pixel limitrofi (ecco perchè c'è l'opzione per scegliere l'algoritmo di interpolazione); l'interpolazione è comunque solo un rimedio parziale che funziona solo in certe parti dell'immagine prive di molti dettagli.

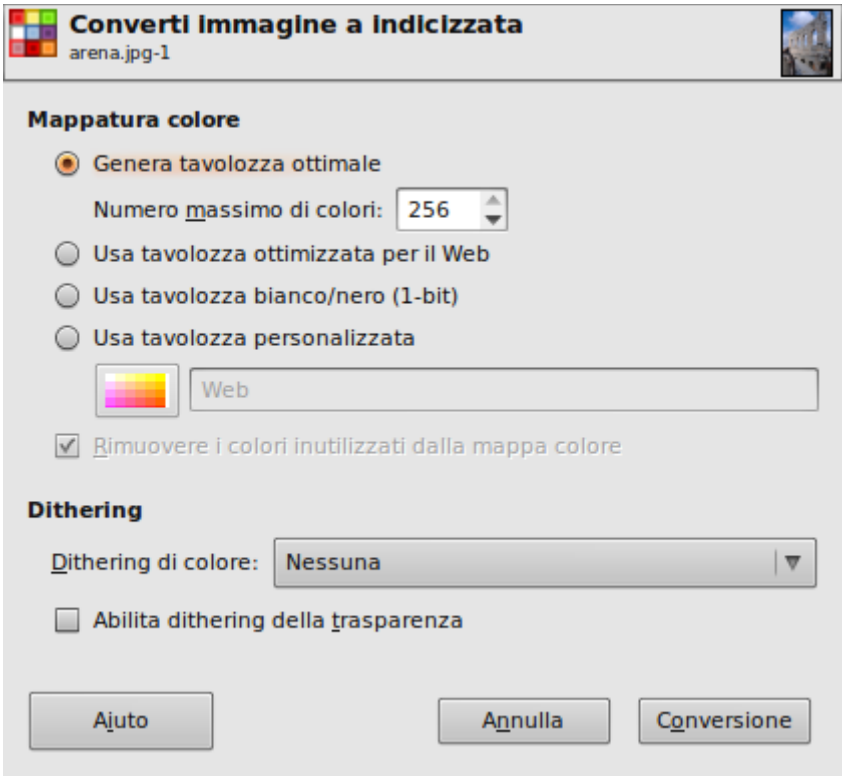
Si noti che un'immagine ridotta di dimensione e poi riportata alla dimensione originale senza mai variare la risoluzione, tornerà ad essere stampata su un'area grande ma con i difetti dovuti alla perdita di informazione.

Questa è l'immagine riportata alla dimensione originale senza alcuna interpolazione. Seppur non fedele a quella mostrata a video, si nota come i bordi degli elementi, e dove ci sono netti contrasti tra i colori, i pixel risultano molto sgranati e i tratti poco definiti.

Esercizio 1: *provare con diverse immagini a ridurre e riprostinare la dimensione specificando diversi tipi di interpolazione per notare le differenze sulla qualità dell'immagine risultante (si salvi sempre in un file a parte la versione ridotta/ripristinata in modo da poterla confrontare con l'originale). Riaprendo tutti i file cosa si può notare ?*

3.3. La profondità di colore

Lavoriamo ora sullo spazio dei colori e sulla profondità dei colori.
Lo spazio colore in cui è rappresentata l'immagine è l'RGB, ovvero un modello additivo basato sui tre colori fondamentali Red Green e Blue.
Dal menu Immagine → Modalità → Scala di grigi possiamo passare allo spazio in scala di grigi così da ottenere l'immagine i bianco e nero.
Possiamo inoltre, con la modalità "Indicizzata" (sempre dal menu Immagine → Modalità →), variare il numero di bit della profondità di colore.



Nel campo "Numero massimo di colori" di default troviamo 256, che corrisponde a 8bit di profondità colore (log base 2 di 256).

Esercizio 2: *provare, sulla stessa immagine di partenza, a variare la profondità di colore a 7, 6 e 4 bit. Si salvi l'immagine trasformata in un secondo file in modo da confrontarla con l'originale. Riaprendo tutti i file cosa si può notare ?*

Cosa accade se si prova, in seguito, a riportare la profondità al valore originale di 8 bit (256 colori) ?

Attenzione! L'operazione di riduzione della profondità di colore di una immagine è irreversibile, in quanto nel momento in cui la si riduce si perde informazione che non si può recuperare anche ritornando alla profondità originale.

3.4. Memorizzazione e compressione delle immagini

La memorizzazione di immagini richiede una grande quantità di spazio.
Facciamo l'esempio di una immagine di 2048x1024 pixel con profondità colore a 32 bit (cioè 8 bit per compomente RGB). Il numero totale di bit necessari a rappresentarne l'informazione è dunque 2048x1024x32, che corrispondono a 8388608 byte.
Questo spazio di memorizzazione può essere ridotto ricorrendo a formati immagine compressi, siano essi lossless che lossy. Con **lossless** si intende una compressione senza perdita di informazione, ovvero consente di decomprimere il file ottenendo gli stessi pixel dell'originale.
Per ottenere una maggior rapporto di compressione si ricorre ad algoritmi **lossy**, ovvero che introduzione perdita di informazione durante il processo di compressione. Uno di questi è adottato dallo standard JPEG.

Il passaggio da immagine Raster al formato JPEG avviene sostanzialmente in tre fasi:

- Utilizzo della trasformata discreta del coseno per la rappresentazione in frequenzia
- Quantizzazione mediante opportune matrici, che conservano maggiormente le basse frequenze spaziali, importanti ai fini della sintesi dell'immagine perchè sono quelle maggiormente percepite del sistema visivo umano
- codica entropica ed eliminazione delle ridondanze di tipo statistico tramite run-length encoding e codici di Huffman

In sostanza vi è un componente di elaborazione volta ad eliminare le informazioni analogiche che il sistema visivo umano percepisce con maggior difficoltà (fase di compressione lossy), e una componente di pura compressione lossless.
Seppur gli algoritmi lossy siano volti ad eliminare le informazioni analogiche che il sistema visivo umano percepisce con maggior difficoltà, se si abusa del fattore di compressione lossy, il degrado della qualità sarà percepibile.

JPEG è lo standard di compressione lossy utilizzato in ambito fotografico, pienamente supportato da GIMP. Altri formati lossless in cui possiamo memorizzare le nostre immagini sono PNG e BMP.

I formati di compressione lossy si utilizzano in applicazioni dove l'utilizzatore finale è un essere umano e l'applicazione non è critica dal punto di vista sanitario; ad esempio si possono usare per le foto delle vacanze o la foto tessera della carta di identità. Le immagini che devono/potrebbero venir analizzate da un sistema informatico non devono essere compresse con formati lossy che potrebbero rimuovere informazioni importanti per l'algoritmo di analisi (ad esempio ricerca automatica di tumori in immagini mediche).

Esercizio 3: *aprire il file dell'Arena di Verona e salvarlo sempre in formato JPEG (basta indicare come estensione .jpg) con valori decrescenti del coefficiente di qualità. Ad esempio salvare in arena60.jpg la versione ottenuta portando la barra della qualità al 60%, passando poi a 30, 15, 10, 1.*

- *Riaprendo tutti i file cosa si può notare ?*
- *Guardando la dimensione dei file cosa si può notare ?*
- *Osservare la dimensione in pixel, risoluzione e profondità di colore di tutti i file; è uguale per tutti ? Si potrebbe dire che il degrado di qualità sia dovuto ad un cambio di uno di questi parametri ?*