

Charts

William Simoni

02 March 2020, Versione: 0.58

Contents

1	Terminologia	2
2	I Grafici, elenco e regole di design	2
2.1	Introduzione	2
2.2	Icon Chart	3
2.3	Gauge Chart	3
2.3.1	Gauge Chart con ago e Compass	3
2.4	Column Chart	4
2.5	Bar Chart	5
2.6	Pie Chart	6
2.6.1	Donut/Doughnut Chart	7
2.7	Line Chart	8
2.8	Area Chart	9
2.9	Stacked Bar/Column Chart	12
2.10	Scatter Plot	13
2.11	Bubble Chart	14
2.12	Histogram	16
2.13	Map	17
3	Un algoritmo magico	18
3.1	Introduzione	18
3.2	L'algoritmo	18
4	Le variabili	23
4.1	Cos'è una variabile	23
4.2	Data Group [non consideratelo]	24
4.3	Funzioni di aggregazione	24
4.4	Interpretare un variable set	25
4.4.1	Variabile semplice	25
4.4.2	Variabile con molti device e keyword	25
4.4.3	Giocare con le funzioni di aggregazione	26

5	Caratteristiche "particolari" dei grafici	32
5.1	Chart Icon	32
5.2	Gauge Chart e Gauge Chart con ago	33
5.3	Column Chart e Bar Chart	34
5.4	Pie Chart e Donut Chart	35
5.5	Line Chart	35
5.6	Stacked Area Chart, 100% Stacked Area Chart, Stacked Bar/- Column Chart, 100 % Stacked Bar/Column Chart	36
5.7	Scatter Plot e Bubble Chart (Forse Non Inclusi)	37
5.8	Histogram	38
6	Fonti	39

1 Terminologia

- Data Source: sorgente del dato. Identificata dalla coppia device e keyword.
- Variabile (variable set): Una variabile è la quadrupla [subtitle, id, dataSources, funzione di aggregazione]. Rappresenta l'informazione identificabile con un id ottenuta applicando una funzione di aggregazione su una o più Data Source.
- Tempo Totale: il periodo di tempo complessivo rappresentato nel grafico.
- Granularità: Valore numerico che indica il numero di volte in cui dividere il Tempo Totale. Quindi indica anche gli intervalli di tempo sui quali applicare le funzioni di aggregazione delle variabili.
- Dato: un dato è la composizione del valore di una, due o tre variabili.
- Data Group: lista di Variabili o Costanti correlate fra loro.
- Label: etichetta informativa che può essere presente all'interno di un grafico per facilitarne la comprensione

2 I Grafici, elenco e regole di design

2.1 Introduzione

In questa sezione intendo elencare i tipi di grafici che andranno a comporre l'output del generatore React. Per ogni grafico vengono anche indicate regole di stile utili per la loro generazione.

2.2 Icon Chart

L'Icon Chart è un semplice grafico che mostra ed evidenzia il valore di una variabile in un determinato momento. L'Icon Chart è caratterizzato da due elementi, l'icona (che deve essere significativa rispetto al valore mostrato) e una parte testuale che mostra il valore della variabile. Per rendere l'Icon Chart più espressivo, potreste usare un'icona costituita da due colori (uno principale e uno secondario), in cui il rapporto tra un valore massimo fissato e il valore della variabile è rappresentato dall'altezza del colore principale (figura 1 a).

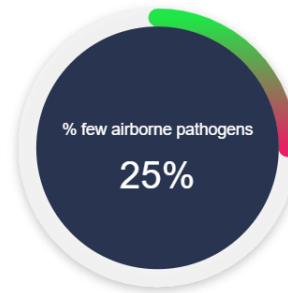
2.3 Gauge Chart

Un Gauge Chart è caratterizzato da un arco circolare e mostra un unico valore di una variabile. Nell'arco circolare viene disegnata una barra progressiva che avanza o arretra a seconda del fatto che il valore mostrato si allontani o si avvicini a una soglia limite fissata.

Usate il Gauge Chart quando volete mostrare quanto manca al raggiungimento di un valore che può essere sia un obiettivo che una soglia di allarme. Inoltre il Gauge Chart può essere usato come l'Icon Chart per evidenziare il valore che una variabile assume in un determinato momento.



(a) Icon Chart



(b) Gauge Chart

Figure 1

2.3.1 Gauge Chart con ago e Compass

Nei Gauge Chart il valore può essere indicato anche tramite l'impiego di un ago che punta al valore attuale della variabile. Una categoria particolare dei Gauge Chart è il **Compass** che rappresenta una bussola. Per esempio, per indicare la direzione del vento attuale possiamo usare il seguente grafico:

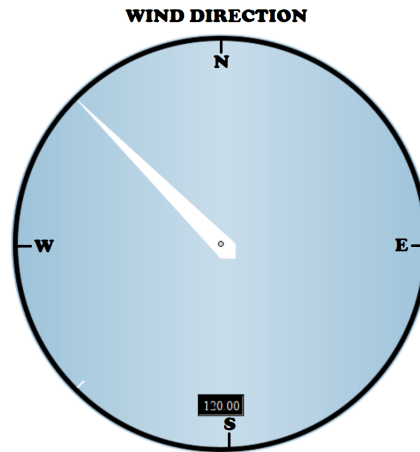


Figure 2: La direzione del vento è indicata da un ago che punta al valore attuale della variabile. (Il grafico mostrato fa abbastanza schifo)

2.4 Column Chart

Il Column Chart è costituito da una serie di colonne verticali disposte lungo un asse orizzontale. Ogni colonna può essere caratterizzata da una label. Viene spesso utilizzato per osservare e confrontare l'andamento **di una o due variabili nel tempo**. Per esempio, possono essere utilizzati per mostrare l'andamento dei guadagni complessivi di un'azienda in un periodo di tempo. Quando le variabili che vogliamo mostrare all'interno del Column Chart sono più di due, allora la visualizzazione rischia di essere confusionaria e il grafico non riesce a raggiungere il suo obiettivo.

Alcune regole di design:

- Usa colori che siano in armonia fra loro. Per le colonne con la stessa data source usa lo stesso colore.
- Usa label orizzontali.
- Fai partire l'asse y da 0.
- La distanza fra le colonne dovrebbe essere più o meno metà della dimensione di una colonna.

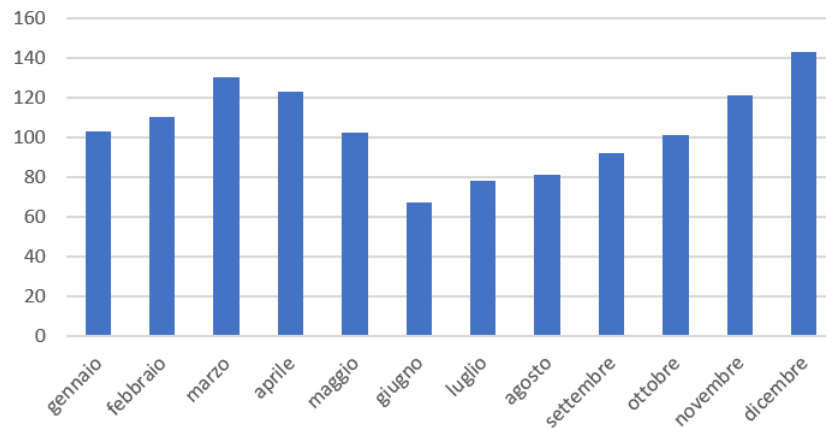


Figure 3: energia elettrica totale prodotta da una centrale idroelettrica per mese durante un anno

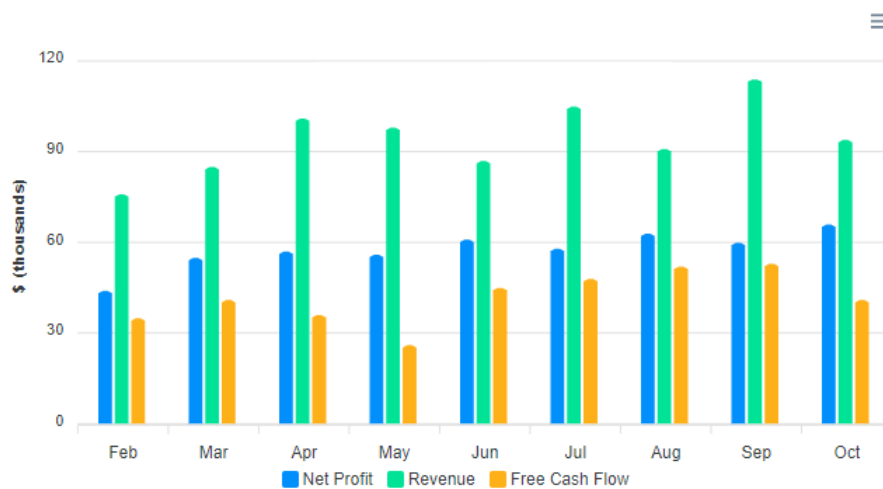


Figure 4: In questo caso invece l'utente non riesce senza difficoltà a capire l'andamento del Net Profit, o della Revenue o del Free Cash Flow

2.5 Bar Chart

Sono simili ai Column Chart, ma le colonne in questo caso sono barre orizzontali disposte lungo un asse verticale. Bar Chart e Column Chart sono abbastanza intercambiabili, tuttavia i Column Chart sono preferibili quando vogliamo confrontare i valori di una o due variabili in vari periodi di tempo, mentre i Bar Chart sono preferibili quando vogliamo confrontare molte variabili in un determinato momento/periodo di tempo. In questo caso il

numero massimo di barre dovrebbe essere quindici. I bar chart sono inoltre consigliabili quando le label sono molte lunghe. Infine sono più efficaci quando è necessario mostrare valori negativi.

Oltre alle regole di design del Column Chart possiamo aggiungere una ulteriore regola:

- Se i valori non devono essere ordinati in qualche modo specifico, allora ordinare le barre dalla più lunga alla più breve (come nell'esempio).

Usare il Bar Chart può essere utile per esempio nel caso vogliate mostrare le 10 nazioni più visitate nel mondo indicando il numero di turisti che le hanno visitate nell'ultimo anno.

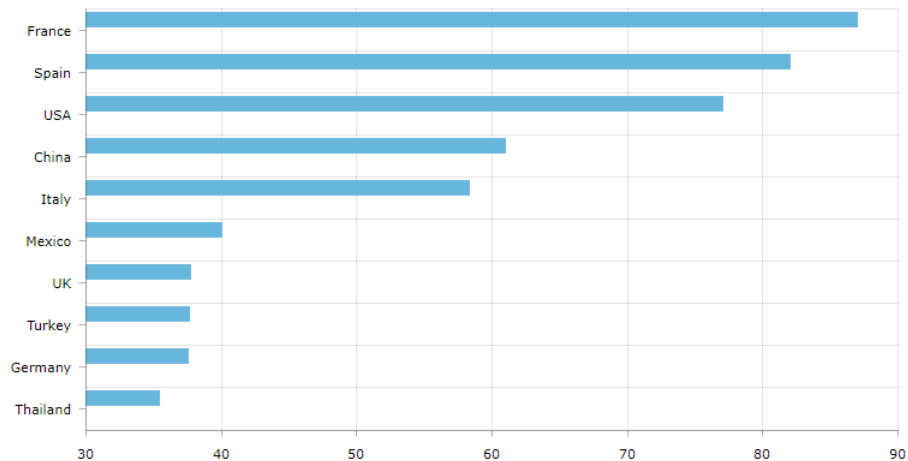


Figure 5: Il grafico mostra le 10 nazioni più visitate nel mondo indicando il numero di turisti in milioni che l'anno scorso hanno visitato la nazione

2.6 Pie Chart

Il Pie Chart è un grafico circolare diviso in fette. Ciascuna delle "fette" rappresenta una delle categorie di dati che insieme costituiscono l'intero/il tutto. L'intera torta rappresenta il 100%. La dimensione di ogni "fetta" è proporzionale alla porzione del tutto che rappresenta.

Il Pie Chart è il modo più semplice per rappresentare come le varie parti di un tutto sono in relazione fra loro **in un determinato periodo** o momento.

Cosa significa? Supponiamo che una compagnia (il tutto) sia composta da cinque divisioni (le parti) e che l'utente voglia sapere quanto ogni divisione abbia contribuito al guadagno totale della compagnia lo scorso anno. Un Pie Chart è perfetto per mostrare questa informazione:

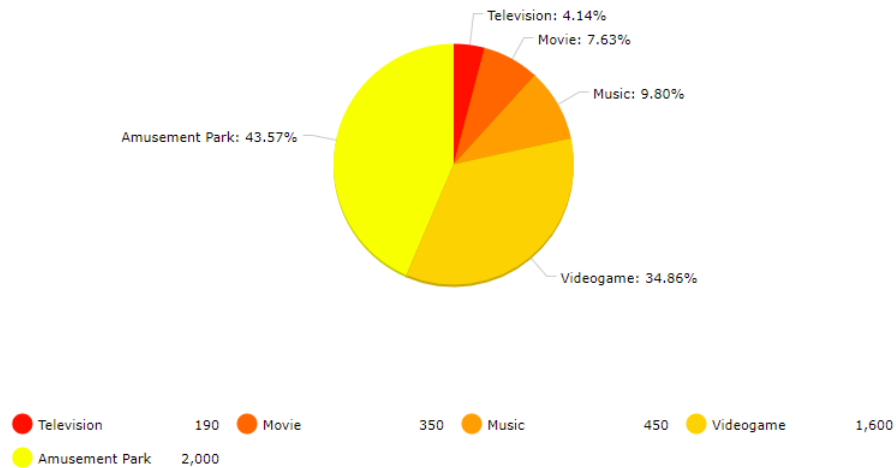


Figure 6: Il Pie Chart permette subito all'utente di capire quali sono le divisioni più redditizie e quelle meno

Quindi generalmente all'interno di un Pie Chart avremmo variabili (le parti) che insieme vanno a comporre un tutto. All'interno di un Pie Chart cercate di mostrare al massimo cinque diverse variabili (possibilmente due). Inoltre fate in modo che all'interno del pie chart siano mostrate tutte le variabili che costituiscono il "Tutto". Altrimenti, il grafico potrebbe trasmettere informazioni sbagliate.

Alcune regole di design:

- La prima fetta deve partire dalle ore 12.
- Ordinate gli spicchi dal più breve al più lungo (come in figura 5).

2.6.1 Donut/Doughnut Chart

Una variante del Pie Chart è il **Donut/Doughnut Chart**. Un Donut Chart è un Pie Chart a forma di ciambella in cui al centro viene mostrata la sommatoria dei valori delle variabili che compongono il grafico. Quindi se l'utente è interessato a conoscere il totale usate un Donut Chart, altrimenti procedete con un Pie Chart.

Alcuni studi hanno dimostrato che alcune persone tendono ad avere un'idea distorta quando vedono un Pie Chart poichè porzioni più grandi possono apparire ancora più grandi. Il problema non persiste con i Donut Chart.

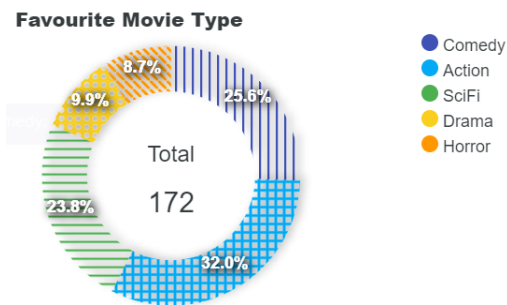


Figure 7: Esempio di Donut. Nell'esempio non viene rispettata la regola "ordinate gli spicchi dal più breve al più lungo"

2.7 Line Chart

Un Line Chart è, come si potrebbe immaginare, una linea o più linee che mostrano come una variabile cambia nel tempo. E' un ottimo strumento per evidenziare l'entità del cambiamento di una o più variabili.

Alcune finzze di design:

- Usa solo linee continue.
- Non superare le quattro righe per evitare che si crei confusione.
- Fai in modo che le linee occupino 2/3 dell'asse y.

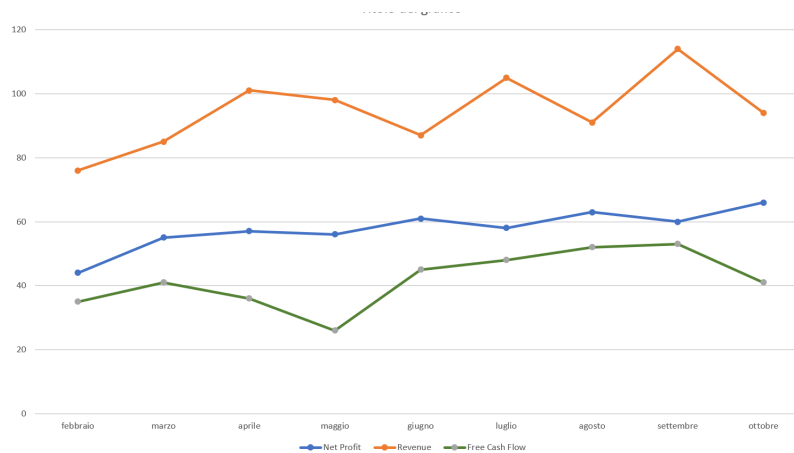


Figure 8: La versione Line Chart del Column Chart in figura 4. L'utente riesce più agilmente a comprendere il trend delle variabili mostrate

Cercate di non usare i Line Chart quando volete mostrare in che misura le

parti di un tutto variano nel tempo. In questo caso la soluzione migliore è un Area Chart.

2.8 Area Chart

Supponiamo di essere l'azienda discussa nell'esempio del Pie Chart e di voler vedere l'andamento delle vendite delle varie divisioni dal 2015 al 2019. Potremmo pensare di rappresentare l'informazione tramite un Column Chart ottenendo questo risultato:

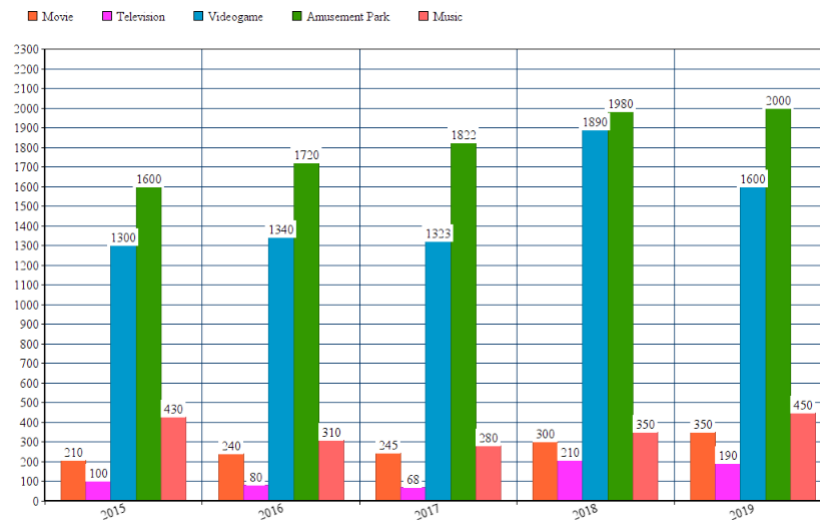


Figure 9: Qualcosa in questo esempio non ci convince

Il problema è che l'utente non riuscirà a capire molto da questo grafico. Come scritto precedentemente non usate il Column Chart con più di due variabili, altrimenti il risultato sarà confusionario e incomprensibile.

Per ottenere un risultato sicuramente più comprensibile e utile possiamo utilizzare un Line Chart con cinque linee:

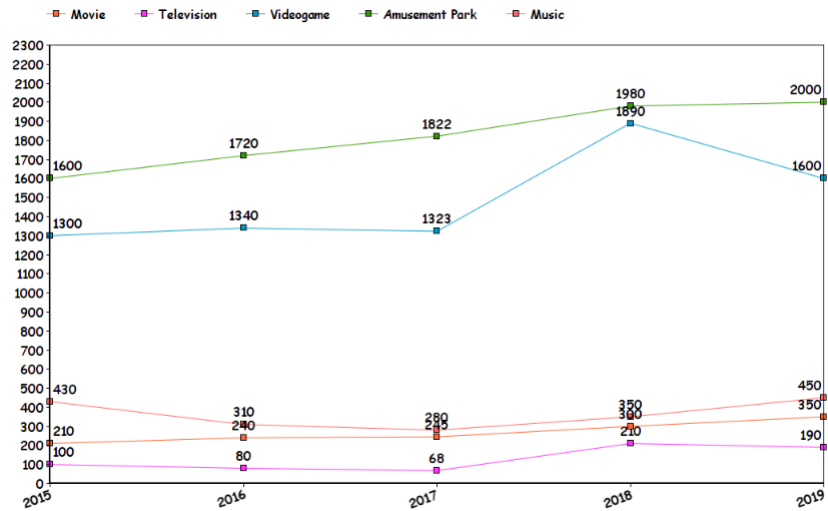


Figure 10: L'utente comprende subito l'andamento delle varie divisioni

Ma come ho scritto prima, quando vogliamo mostrare come le parti di un tutto cambiano nel tempo, bisogna cercare di evitare i line chart.

L'Area Chart è sicuramente una soluzione migliore. L'Area Chart è un Line Chart in cui viene colorata l'area sotto la linea. Esistono tre tipi di Area Chart, il Regular, lo Stacked e il 100% Stacked.

Ogni volta che vogliamo mostrare come le parti di un tutto variano nel tempo dovremmo pensare a un Area Chart.

Se siamo interessati a evidenziare una tendenza dovremmo usare uno **Stacked Area Chart**. Negli Stacked Area Chart, i valori di ogni variabile sono mostrati l'uno sopra l'altro (Le linee non si intersecano mai).

Lo Stacked Area Chart consente di osservare contemporaneamente sia in che misura le parti contribuiscono alla composizione, ma anche come l'intera composizione varia nel tempo. Usate invece un 100% Stacked Area Chart se siete interessati a evidenziare in che misura ogni parte della composizione contribuisce alla composizione totale.

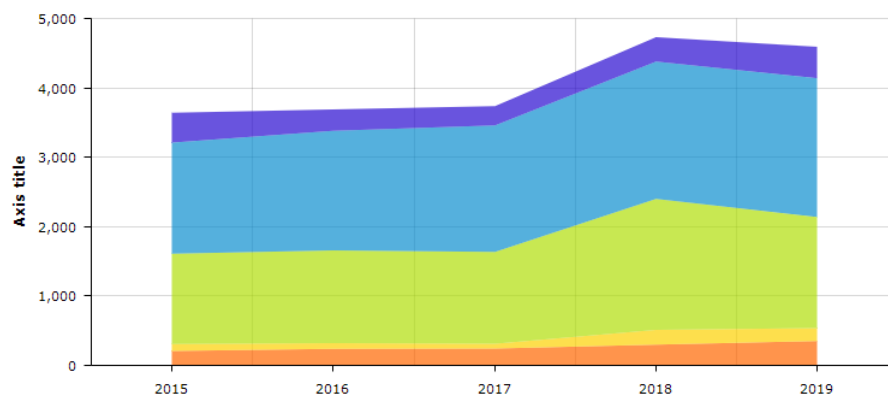


Figure 11: Stacked Area Chart. Dal grafico possiamo captare facilmente tre informazioni, l'andamento della compagnia, l'andamento delle varie divisioni che compongono la compagnia e infine anche il guadagno totale. Come potete vedere le linee non si intersecano fra loro, ma sono l'una sopra l'altra.

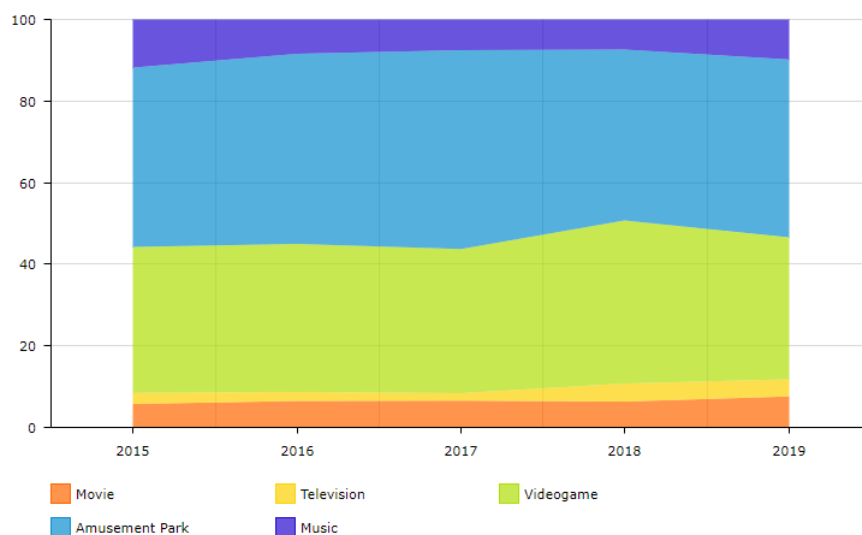


Figure 12: 100% Stacked Area Chart. In questo caso invece evidenzio l'importanza di ogni divisione nel tempo

Non usate l'Area Chart per rappresentare in che misura le parti contribuiscono al tutto in un determinato periodo o momento (per quello c'è il Pie Chart, il Donut Chart o gli Stacked Column/Bar Chart). Cercate di evitare anche il Regular Area Chart. Se infatti usassimo il Regular Area Chart per

rappresentare l'andamento delle varie divisioni della compagnia, otterremmo una schifezza del genere:

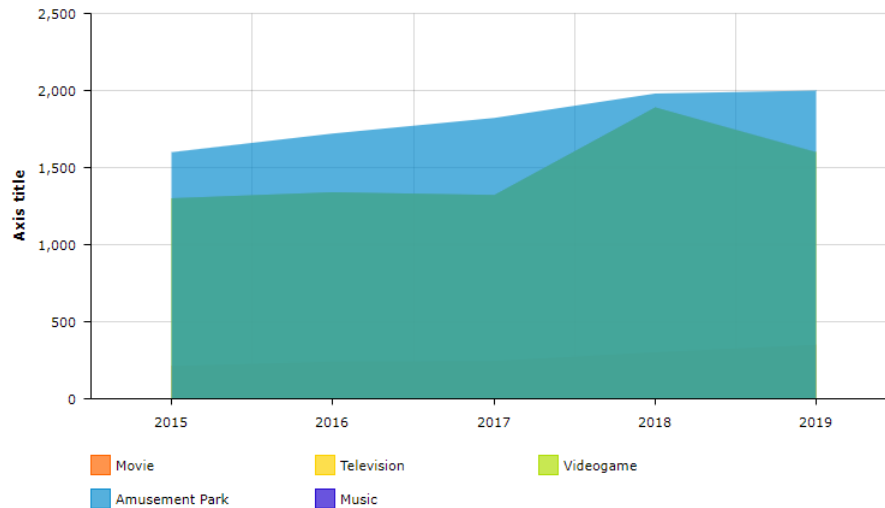


Figure 13: Nessun commento, una vera e propria schifezza

Evitate anche gli Stacked Line Chart e 100% Line Chart. Alcuni consigli di stile per gli Area Chart:

- Usa colori trasparenti
- Inserisci le data source più variabili in alto
- Non superare le cinque categorie.

2.9 Stacked Bar/Column Chart

Lo Stacked Bar/Column Chart è ottimo quando volete mostrare più composizioni di variabili in un determinato momento. Cercate di non avere composizioni composte da troppe variabili. Vale la solita regola dei Stacked Area Chart e 100% Stacked Area Chart; se volete mostrare sia in che misura le parti contribuiscono alla composizione totale, ma anche mostrare il valore della intera composizione usate uno Stacked Bar/Column Chart, se invece volete evidenziare in che misura ogni parte della composizione contribuisce sulla intera composizione usate un 100% Stacked Chart.

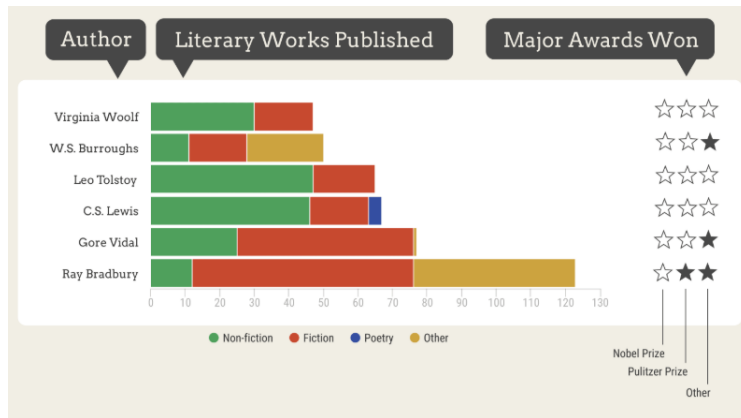


Figure 14: Uno Stacked Bar Chart di venngage. Vengono mostrate contemporaneamente 6 composizioni, una per ogni scrittore

2.10 Scatter Plot

Se volete mostrare la relazione tra due variabile il modo migliore di farlo è tramite uno Scatter Plot. Uno Scatter Plot è un grafico in cui i singoli valori delle variabili sono riportati su uno spazio cartesiano.

Per esempio, supponiamo di essere un'azienda di trasporti pubblici e di voler osservare la relazione che c'è tra i chilometri percorsi da un mezzo in un giorno e il costo per far funzionare il mezzo nel medesimo giorno (rifornimento del carburante). Inoltre, la nostra azienda possiede vari mezzi che funzionano tramite l'impiego di vari carburanti che sono rispettivamente un carburante arancione e uno verde. Quindi nel grafico vorremmo poter capire se stiamo osservando il dato di un mezzo che va ad arancione o di un mezzo che va a verde.

In questo caso il dato che vogliamo mostrare potrebbe essere il seguente:

- x: la variabile "chilometri percorsi dal mezzo nel giorno z".
- y: la variabile "costo per far funzionare il mezzo nel giorno z".

Dove z assume i valori da ieri fino a 1 settimana fa.

Il grafico migliore per rappresentare tale dato è sicuramente uno scatter plot:

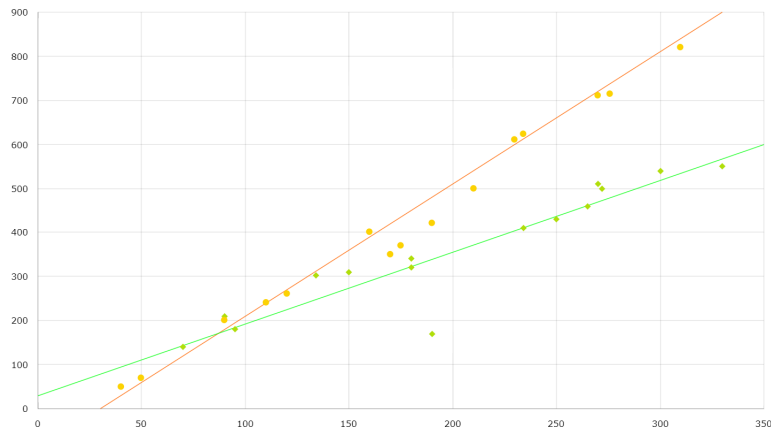


Figure 15: Sull'asse x abbiamo indicato i chilometri percorsi dal mezzo nel giorno z, mentre sulla y il costo per farlo funzionare nel medesimo giorno. Nota che l'utente può subito captare tre informazioni: 1) più sono i chilometri, maggiore è il costo del mezzo, 2) i veicoli del carburante arancione costano meno quando i chilometri percorsi sono pochi (circa meno di 100 km) mentre costano di più quando i chilometri percorsi sono molti, 3) C'è un outlier (in italiano un dato anomalo) per quanto riguarda i veicoli con il carburante verde, infatti un punto è molto distante dalla linea che mostra il trend del carburante verde

Ovviamente lo Scatter Plot è utile quando le due variabili sono logicamente correlate. Nel grafico sopra, si dice che le variabili sono correlate positivamente perché lo sciame di punti tende verso l'alto. Se lo sciame avesse teso verso il basso, allora le variabili sarebbero state correlate negativamente. Se le variabili non sono correlate, il risultato sarà un insieme di punti disposti casualmente all'interno dell'asse cartesiano.

Non utilizzate lo Scatter Plot quando volete osservare l'andamento nel tempo delle variabili. Infine, lo Scatter Plot ha senso solo se le variabili sono valori numerici.

Alcune regole di design:

- Fate partire l'asse y da 0
- Usate al massimo due linee per mostrare il trend dello sciame di punti.

2.11 Bubble Chart

Torniamo all'azienda dell'esempio precedente. Oltre alle due informazioni già descritte, vogliamo mostrare anche quanto ogni autobus ha guadagnato nel giorno z. Questo terzo dato lo possiamo mostrare usando la dimensione del punto nel piano cartesiano. Più è grande il punto più è stato alto il guadagno.

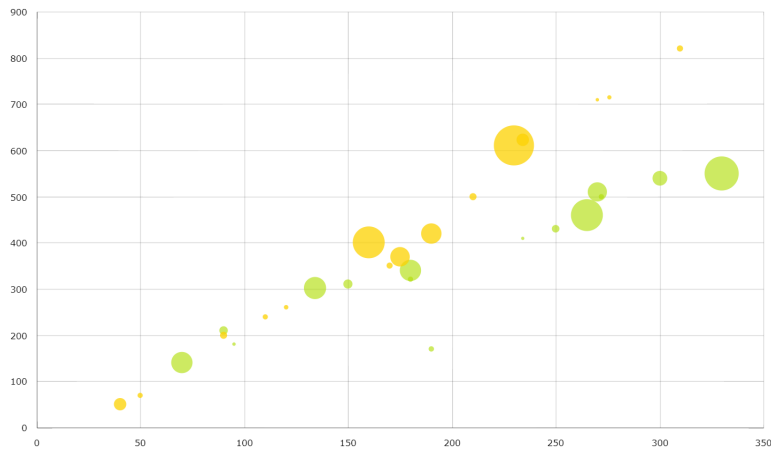


Figure 16: In questo modo osservo la relazione tra tre variabili correlate

Il grafico mostrato si chiama Bubble Chart e può essere utilizzato quando vogliamo mostrare la relazione tra tre variabili in qualche modo correlate. Quindi ai due assi vengono associate due variabili mentre la terza variabile è associata alla dimensione del punto. I Bubble Chart come li Scatter Plot permettono all'utente di trovare velocemente dati anomali.

Vediamo un'altra applicazione del Bubble Chart: abbiamo posizionato, all'interno di un parco naturale dei sensori di prossimità che inviano un allarme ogni qual volta qualcosa si trova nelle immediate vicinanze, presumibilmente animali. Ogni volta che il sensore rivela un movimento invia un segnale a un server il quale memorizza l'ora dello spostamento e il device che ha inviato il segnale. Vogliamo vedere, tramite una mappa, i punti in cui sono stati rilevati più movimenti e i punti in cui ne sono stati rilevati meno negli ultimi sette giorni in modo da mappare, in qualche modo, gli spostamenti degli animali all'interno del parco.

All'asse X e all'asse Y associamo la posizione dei device sulla mappa (che conosciamo ed è una Costante). La dimensione del punto sarà invece associata alla variabile "numero di allarmi per il device con id #".

Il grafico finale potrebbe essere questo:

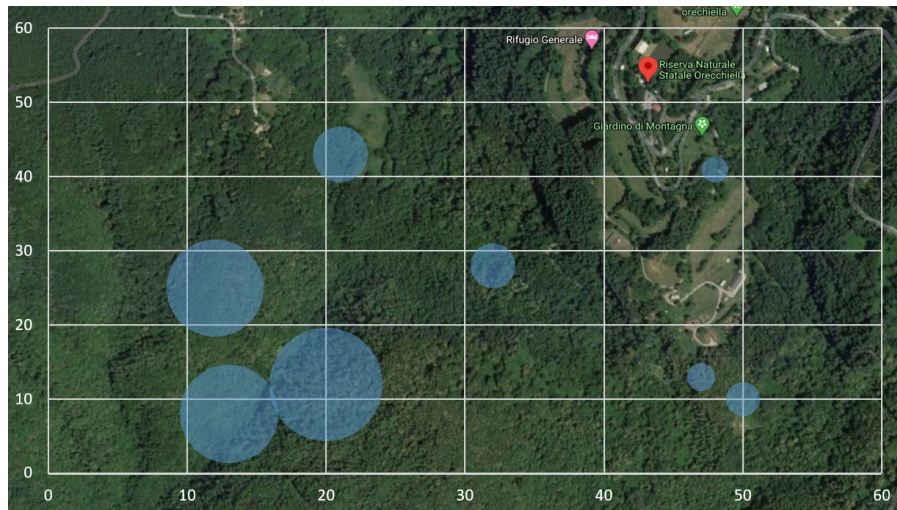


Figure 17: La mappa mostra i luoghi del parco più colpiti dagli animali

Alcuni consigli di stile:

- Usa solo cerchi
- Fai in modo che il colore di riempimento dei cerchi sia semi trasparente.

2.12 Histogram

Come scritto su wikipedia "L'istogramma è la rappresentazione grafica (diagramma) di una distribuzione in classi di un carattere continuo". Possiamo usare un Histogram per mostrare la **distribuzione dei valori** assunti **da una variabile**. (Per mostrare la distribuzione di due variabili correlate usate uno Scatter Plot).

Consideriamo questo esempio. Vogliamo mostrare la distribuzione della variabile: "temperatura media oraria di un dispositivo" durante la giornata di ieri.

Per farlo utilizziamo un Histogram composto dalle seguenti classi:

- temperature inferiore a 0
- temperatura compresa tra 0 e 14
- temperatura compresa tra 15 e 29
- temperatura compresa tra 30 e 44
- temperatura compresa tra 45 e 59
- temperatura compresa tra 60 e 74

- temperatura maggiore di 75

Durante un giorno, la variabile "temperatura media oraria di un dispositivo" assume 24 valori. Otteniamo quindi il seguente Histogram:

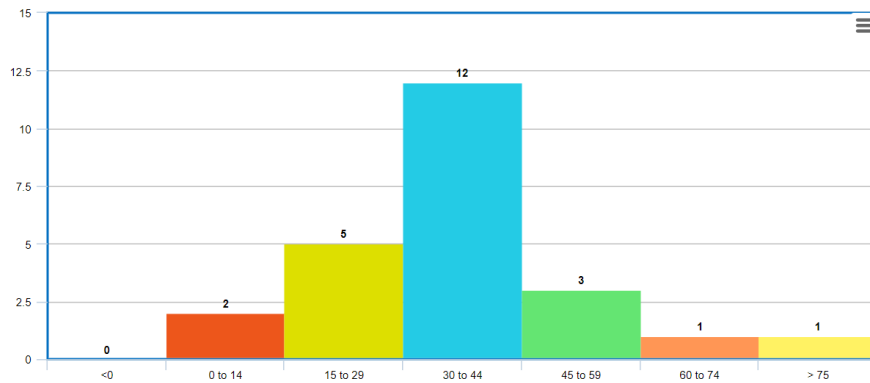


Figure 18: Il grafico ci dice, che nella giornata di ieri, il device ha registrato una temperatura minore di 0 gradi 0 volte, fra 0 e 14 gradi 2 volte ecc.

Non utilizzate gli Histogram per mostrare la distribuzione di dati che sono composti da più di una variabile.

Valgono le stesse regole di stile del Column Chart eccetto per il fatto che le colonne possono anche essere attaccate fra loro.

2.13 Map

Le mappe sono ottime quando volete mostrare i dati geografici in base alla posizione. I dati in una mappa sono spesso mostrati tramite mappe ad aree colorate o Bubble Chart (come mostrato sopra).

Potete usare le mappe per mostrare la posizione in tempo reale di dispositivi mobili. Per esempio, potreste usare una mappa per mostrare dove si trova la ISS in questo momento (<https://wheretheiss.at/>).

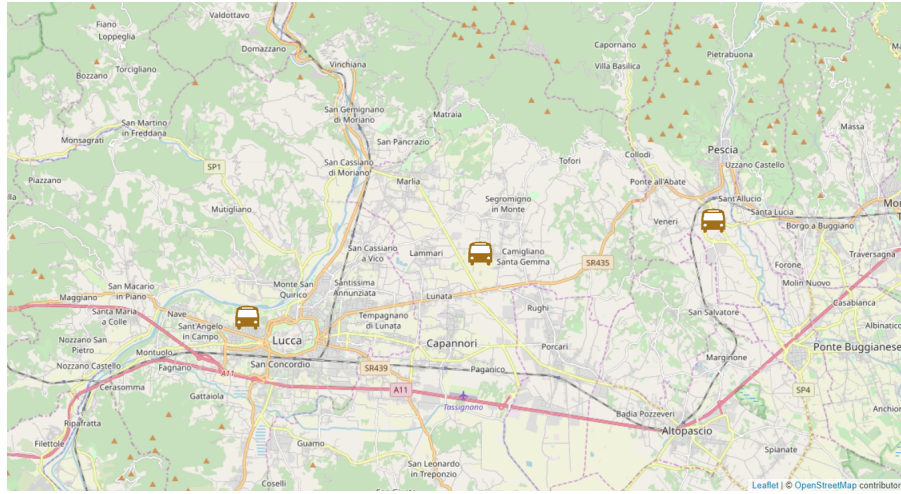


Figure 19: La mappa mostra lo stato e la posizione dei alcuni autobus. Mappa realizzata con leaflet.js e con le mappe di OpenStreetMap

3 Un algoritmo magico

3.1 Introduzione

In questa sezione proverò a descrivere un algoritmo che aiuti una persona a capire che grafico generare in base alle sue esigenze.

3.2 L'algoritmo

Il capitolo proporrà una serie di domande a risposta multipla. Ogni domanda è contrassegnata da un numero. Una volta che avete risposto passate alla domanda con il numero indicato nella risposta. Partite dalla domanda 1. L'algoritmo termina quando arrivare a un numero con la parola chiave TERMINAZIONE.

1) Che informazione vuoi trasmettere all'utente?

- (a) Voglio evidenziare un singolo valore. (2)
- (b) Voglio mostrare differenze o affinità tra i valori di una o più variabili. (3)
- (c) Voglio mostrare la composizione di varie variabili. (10)
- (d) Voglio mostrare l'andamento nel tempo (e quindi il trend) di una o più variabili. (23)
- (e) Voglio mostrare le relazioni che intercorrono tra varie variabili. (24)

(f) Voglio mostrare la distribuzione dei valori di una o più variabili. (28)

2) terminazione

Dato che volete evidenziare il singolo valore assunto da una variabile i due grafici che puoi utilizzare sono:

- Icon Chart
- Gauge Chart

3) Come intendi confrontare i valori delle variabili?

- (a) in base a vari periodi o momenti temporali. (4)
- (b) in base a un singolo periodo o momento temporale. (7)

4) Quante sono le variabili

- (a) una o due. (5)
- (b) più di due. (6)

5) Quanti sono i periodi/momenti?

- (a) Pochi (meno di 7/8) (11)
- (b) molti (12)

6) terminazione

Vuoi mostrare differenze o affinità tra i valori di varie variabili indipendenti in vari momenti o periodi. Questo significa che il grafico più adatto è un Line Chart.

7) Quante sono le variabili?

- (a) una o due. (8)
- (b) più di due. (9)

8) terminazione

Vuoi mostrare differenze o affinità tra i valori di una o due variabili indipendenti in un determinato momento o periodo. Questo significa che il grafico più adatto è un Column Chart.

9) terminazione

Vuoi mostrare differenze o affinità tra i valori di varie variabili indipendenti un determinato momento o periodo. Questo significa che il grafico più adatto è un Bar Chart.

10) Come vuoi che sia la composizione?

- (a) Composizione delle variabili in un determinato momento o periodo. (13)
- (b) Composizione delle variabili in vari momenti o periodi (16)

11) terminazione

Vuoi mostrare differenze o affinità tra i valori di varie variabili indipendenti in pochi momenti o periodi. Questo significa che il grafico più adatto è un Column Chart.

12) terminazione

Vuoi mostrare differenze o affinità tra i valori di varie variabili indipendenti in molti momenti o periodi. Questo significa che il grafico più adatto è un Line Chart.

13) Quante sono le composizioni che vuoi mostrare?

- (a) una. (14)
- (b) più di una (15)

14) terminazione

Vuoi mostrare un unica composizioni di variabili in un determinato momento. Il grafico più adatto è un Pie Chart.

15) terminazione

Vuoi mostrare più composizioni di variabili in un determinato momento. Il grafico più adatto è uno 100% Stacked Bar Chart o uno 100% Stacked Column Chart.

16) Quanti sono i periodi/momenti?

- (a) Pochi (massimo 3/4) (17)
- (b) molti (20)

17) Cosa vuoi mostrare nello specifico?

- (a) Voglio evidenziare come ogni parte della composizione contribuisce sull'intera composizione(18)
- (b) Voglio mostrare sia come le parti contribuiscono alla composizione, ma anche come l'intera composizione varia nel tempo(19)

18) terminazione

Vuoi mostrare la composizione di variabili in pochi momenti evidenziando come ogni parte della composizione contribuisce sull'intera composizione. Il grafico migliore è un 100% Stacked Column Chart o 100% Stacked Area Chart.

19) terminazione

Vuoi mostrare la composizione di variabili in pochi momenti non solo mostrando come le parti contribuiscono alla composizione, ma anche come l'intera composizione varia nel tempo. Il grafico migliore è uno Stacked Column Chart o Stacked Area Chart.

20) Cosa vuoi mostrare nello specifico?

- (a) Voglio evidenziare come ogni parte della composizione contribuisce sull'intera composizione(21)
- (b) Voglio mostrare sia come le parti contribuiscono alla composizione, ma anche come l'intera composizione varia nel tempo(22)

21) terminazione

Vuoi mostrare la composizione di variabili in molti momenti evidenziando come ogni parte della composizione contribuisca sull'intera composizione. Il grafico migliore è un 100% Stacked Area Chart.

22) terminazione

Vuoi mostrare la composizione di variabili in molti momenti evidenziando come ogni parte della composizione contribuisca sull'intera composizione. Il grafico migliore è uno Stacked Area Chart.

23) Quanti sono i periodi/momenti?

- (a) Pochi (meno di 7/8) (31)
- (b) molti (32)

24) Come sono relazionate le variabili?

- (a) in coppia. (25)
- (b) in gruppi di tre. (26)
- (c) più di tre. (27)

25) terminazione

Vuoi mostrare le relazioni che intercorrono fra i valori di due variabili. Il grafico per queste evenienze è un Scatter Plot.

26) terminazione

Vuoi mostrare le relazioni che intercorrono fra i valori di tre variabili. Il grafico per queste evenienze è un Bubble Chart.

27) terminazione

Vuoi mostrare le relazioni che intercorrono fra i valori di tante variabili. Non lo so.

28) Quante sono le variabili?

- (a) una. (29)

(b) due. (30)

29) terminazione

Mostrare la distribuzione di una variabile? Nulla di più semplice con un Histogram

30) terminazione

Vuoi mostrare la distribuzione di due variabili. Quindi dovresti usare uno Scatter Plot.

31) Quante sono le variabili?

(a) una o due. (32)

(b) più di due. (33)

32) terminazione

Vuoi confrontare i valori di poche variabili in pochi periodi di tempo. Per cui ti consiglio un semplice Column Chart.

33) terminazione

Vuoi confrontare i valori di molte variabili in pochi periodi di tempo. Per cui ti consiglio un Line Chart.

34) terminazione

Vuoi confrontare i valori di alcune variabili in molti periodi di tempo. Per cui ti consiglio un Line Chart.

4 Le variabili

4.1 Cos'è una variabile

Avete scelto il grafico perfetto per voi! (A meno che non siate finiti al numero 27...) Per ogni grafico dovrete anche indicare il periodo totale che viene rappresentato nel grafico (un anno, un giorno ecc.) e la granularità del grafico (1,2,3,ecc.). Inoltre alcune categorie di grafici hanno proprietà particolari che saranno oggetto del prossimo capitolo.

Ora dovete definire le variabili che popoleranno il grafico. Le variabili vengono definite tramite dei **variable set**. Ogni variable set ha quattro proprietà:

- **subtitle**: nome che indica, tramite un nome significativo, il nome dei device indicati in data source.
- **id**: id numerico della variabile generato automaticamente.
- **data sources**: composta da un array di nomi di device e un array di nomi di keyword. L'insieme di tutte le data source è l'insieme di tutte le coppie ordinate (nome del device, nome della keyword).
- **funzione di aggregazione**: Una key word che indica la funzione da applicare sui dati inviati dalle/a data source.

4.2 Data Group [non consideratelo]

Se volete creare uno Scatter Plot, un Bubble Chart, uno Stacked Column/Bar Chart o un grafico di composizione (Pie Chart, Stacked Area Chart, ecc.) dovete inoltre indicare come sono organizzate le variabili. In altre parole dovete definire dei **Data Group**. Le variabili inserite all'interno del medesimo Data Group saranno o un punto nei primi due grafici, o una composizione negli ultimi due.

Ogni Data Group è caratterizzato da due proprietà:

- **name**: il nome del gruppo. Non è obbligatorio inserirlo.
- **group**: un Data Group che può essere di default o personalizzato.

Quando viene creata una variabile (con funzione "per device") implicitamente vengono creati quanti Data Group quanti sono i device definiti nella variabile. Ognuno di questi Data Group sarà costituito da tutte le coppie (device, keyword) per ogni keyword definita.

Quando viene creata una variabile (con funzione "sui device") implicitamente viene creato un unico Data Group per tutti i device definiti nella variabile.

Questo Data Group sarà costituito da tanti elementi quante sono le keyword. E' possibile inoltre definire Data Group personalizzati. I Data Group personalizzati possono contenere variabili con un solo device e una sola keyword oppure costanti.

Per comprendere meglio il concetto leggete i prossimi esempi.

4.3 Funzioni di aggregazione

Ogni funzione di aggregazione è caratterizzata da tre proprietà:

- **type**: stringa che identifica il tipo della funzione di aggregazione. Per esempio sum, min o max.

- **aggregated**: un array che può essere costituito dalle parole "device" e "keyword". Se l'array contiene "device", allora "device" non può essere inserito in **divided**. Lo stesso vale per "keyword".
- **divided**: un array che può essere costituito dalle parole "device" e "keyword". Se l'array contiene "device", allora "device" non può essere inserito in **aggregated**. Lo stesso vale per "keyword".

Nel prossimo capitolo verranno proposti degli esempi per farvi capire come utilizzare le proprietà aggregated e divided.

4.4 Interpretare un variable set

La definizione dei variable set è fondamentale nella generazione di un grafico. Per questo è importantissimo comprenderle in modo da poterle utilizzare al massimo delle possibilità. In questa sottosezione, elencherò degli esempi di variabili seguiti dalla loro "interpretazione testuale".

4.4.1 Variabile semplice

- subtitle: "Caldaia"
- id: variable1
- data Source:
 - device: [device1]
 - keyword: [Temperatura]
- funzione di aggregazione:
 - type: max
 - aggregated: []
 - divided: [device, keyword]

Tecnicamente è la variabile ottenuta applicando la funzione "massimo" sui dati ottenuti a partire dalla data source [device1, Temperatura]. Tale dato è descritto dalla frase "Temperatura Caldaia".

In questo caso l'unica proprietà significativa della funzione di aggregazione è type. Questo perché il variable set è costituito da una singola variabile.

4.4.2 Variabile con molti device e keyword

- subtitle: ["Caldaia gialla", "Caldaia zona A", "Caldaia rossa", "Reattore zona A", "Reattore zona B", "Termometro frigorifero di casa"]
- id: variable2
- data Source:

- device: [AT324, AT314, AT334, AT344, AT354, AT364]
- keyword: [Temperatura, Pressione, Umidità]
- funzione di aggregazione: ...

In questo caso è come se stessimo definendo all'interno di un'unica variabile 18 variabili diverse, una per ogni coppia [device, keyword].

4.4.3 Giocare con le funzioni di aggregazione

In questi esempi vorrei porre l'accento su come è possibile sfruttare le proprietà aggregated e divided delle funzioni di aggregazione. Supponiamo di essere i proprietari di un bar in cui i baristi sono quattro robot che, per ogni bevanda prodotta e venduta, inviano al server alcuni dati. Abbiamo deciso di dividere le bevande in quattro categorie:

- Bevanda analcolica
- Bevanda analcolica a base di latte
- Bevanda calda
- Bevanda alcolica

Vogliamo mostrare, in tempo reale, il numero di bevande prodotte e servite da ogni robot, mostrando anche la suddivisione sopra citata nelle ultime 24 ore. Il modo migliore per mostrare questo dato è tramite uno Stacked Bar Chart con tempo totale 24 ore e granularità 1.

Definiamo quindi la variabile che ci permetterà di ottenere i dati:

- subtitle: ["Salvatore Calabrese", "Francesco Cione", "Matteo 'Zed' Zamberlan", "Luca Cinalli"]
- id: var1
- data source:
 - device: ["ROBOT1", "ROBOT2", "ROBOT3", "ROBOT4"]
 - keyword: ["bevanda analcolica", bevanda analcolica a base di latte", "bevanda calda", "bevanda analcolica"]
- funzione di aggregazione:
 - type: somma
 - aggregated: []
 - divided: [device, keyword]

Definendo la funzione di aggregazione in questo modo non vengono aggregati né i risultati sui device né i risultati sulle keyword. Il risultato è un set di 16 variabili, una per ogni coppia ordinata (nome del device, keyword). Su ognuna di queste variabili viene applicata la funzione somma. In altre parole ogni variabile restituisce il valore ritornato dalla seguente funzione (in modo molto semplificato):

```
1 function (deviceName, keyword)
2 return Somma(device.keyword);
3 # Somma e' la funzione di aggregazione applicata sui dati
4 # inviati dalla data source identificata
5 # dalla coppia (deviceName, keyword).
```

Infine dato che il grafico è di composizione, i dati vengono organizzati in base al device.

Otteniamo quindi il seguente grafico:

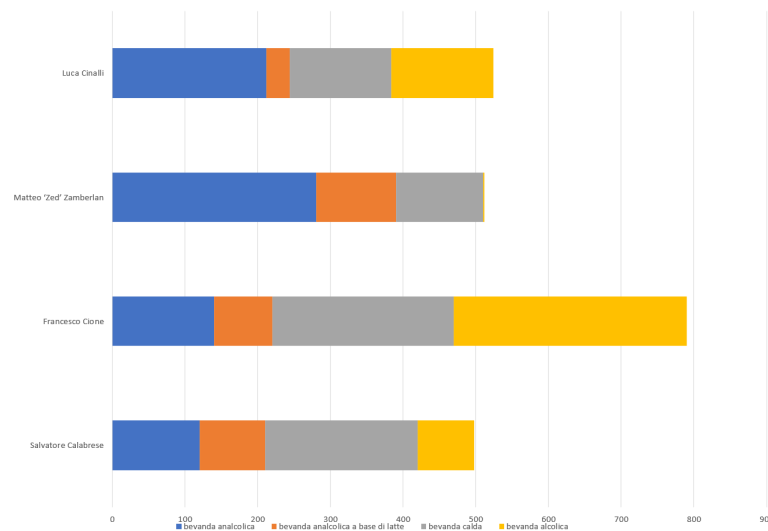


Figure 20: Stacked Bar Chart

Proviamo a definire la funzione di aggregazione scambiando le parole device e keyword in divided:

- type: somma
- aggregated: []
- divided: [keyword, device]

In questo caso otteniamo questo grafico:

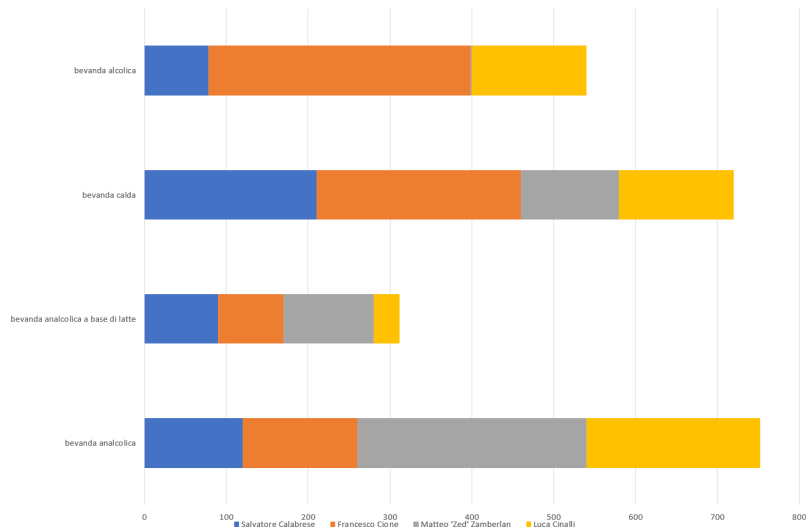


Figure 21: Stacked Bar Chart

Quindi l'ordine delle parole device e keyword può essere significativo nei grafici di composizione come lo Stacked Bar Chart. Se viene scritta prima la parola device, allora la composizione avviene in base al device, viceversa, avviene in base alla keyword.

Supponiamo che l'azienda si sia espansa e che ora possieda due bar. Il nuovo bar è dotato di quattro robot di ultima generazione che secondo il venditore sono più veloci e gentili rispetto ai robot installati nel primo bar. Vogliamo capire se quello che afferma il venditore è vero e quindi mostrare il numero totale di bevande vendute dai due bar.

In questo caso dobbiamo definire due variabili. Dato che sono praticamente identiche, scrivo solo quella relativa al primo bar:

- subtitle: ["Salvatore Calabrese", "Francesco Cione", "Matteo 'Zed' Zamberlan", "Luca Cinalli"]
- id: bar1
- data source:
 - device: ["ROBOT1", "ROBOT2", "ROBOT3", "ROBOT4"]
 - keyword: ["bevanda analcolica", "bevanda analcolica a base di latte", "bevanda calda", "bevanda analcolica"]
- funzione di aggregazione:
 - type: somma
 - aggregated: [device, keyword]

– divided: []

Notate che ho scritto le parole "device" e "keyword" nella proprietà aggregated della funzione di aggregazione.

Il grafico in questo caso sarà un Pie Chart con tempo totale di 1 mese e granularità 1:

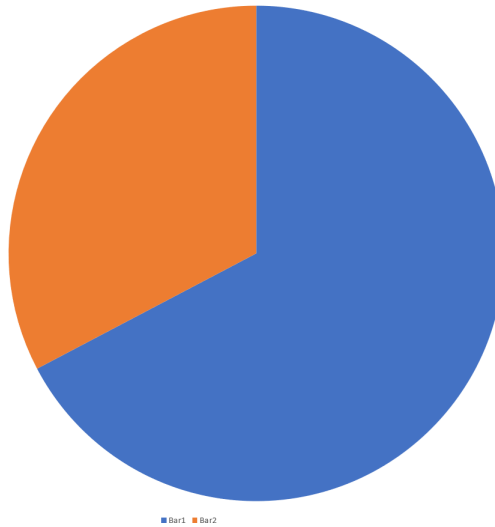


Figure 22: Sicuramente non è il risultato atteso

Quindi inserendo device e keyword in aggregated, le somme ottenute sulle singole variabili (device, keyword) vengono nuovamente sommate per ottenere una somma totale. In altre parole, la variabile restituisce il valore ritornato da questa funzione:

```
1  function ()
2      somma = 0
3      for (device in datasources.device)
4          for (keyword in datasources.keyword)
5              somma += device.keyword
6      return somma;
```

Cambiare l'ordine delle parole nella proprietà aggregated non cambia il risultato.

Dopo mesi di osservazioni, il risultato non cambia; il bar 1 produce più bevande del bar 2. I robot di nuova generazione non sembrano un granché. Ma un giorno ci accorgiamo che ai robot è stato impostato un livello di gentilezza pari a 100 su 100, decidiamo quindi di sperimentare. Al robot chiamato "Dario Comini" assegniamo una gentilezza pari a 25, a "Domenico Costa" assegniamo una gentilezza di 50, a "Luca Picchi" assegniamo 75, infine a "Peppino Manzi"

assegniamo 100. Vogliamo mostrare il numero di bevande prodotte e vendute dai quattro robot nell'ultimo mese con le gentilezze così assegnate.

Definiamo la seguente variabile:

- subtitle: ["Dario Comini", "Domenico Costa", "Luca Picchi", "Peppino Manzi"]
- id: bar2
- data source:
 - device: ["ROBOT5", "ROBOT6", "ROBOT7", "ROBOT8"]
 - keyword: ["bevanda analcolica", bevanda analcolica a base di latte", "bevanda calda", "bevanda analcolica"]
- funzione di aggregazione:
 - type: somma
 - aggregated: [keyword]
 - divided: [device]

Notate che nella funzione di aggregazione, ad aggregated abbiamo assegnato keyword mentre a divided abbiamo assegnato device. In questo caso si ottengono quattro variabili (una per device) ognuna delle quali restituisce il valore ritornato dalla seguente funzione (in modo molto semplificato):

```
1  function (deviceName)
2  somma = 0
3  for (keyword in datasources.keywords)
4      somma += Somma(device.keyword)
5  return somma;
```

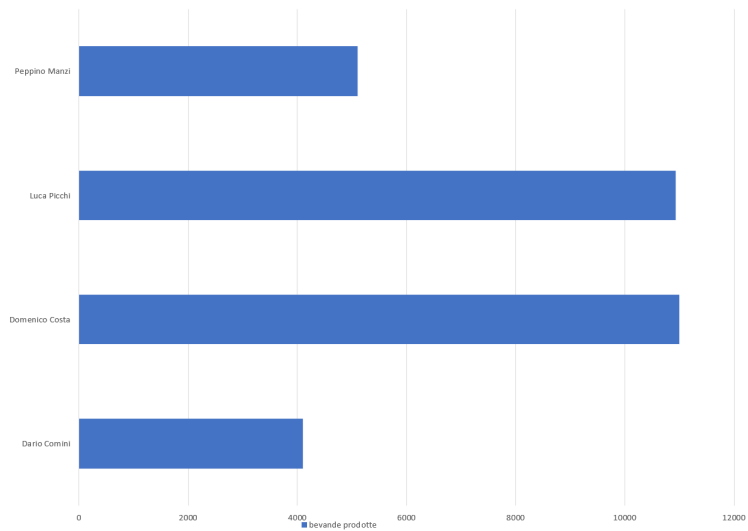


Figure 23

A quanto pare il problema era il livello di gentilezza. Osservazioni sul campo hanno inoltre mostrato che, se è poco gentile il robot offende i clienti e lancia le bevande prodotte, mentre se troppo gentile il robot regala le bevande. In tutti e due i casi la bevanda non veniva venduta e quindi il server non veniva notificato.

Infine proviamo l'ultimo caso di funzione di aggregazione:

- type: somma
- aggregated: [device]
- divided: [keyword]

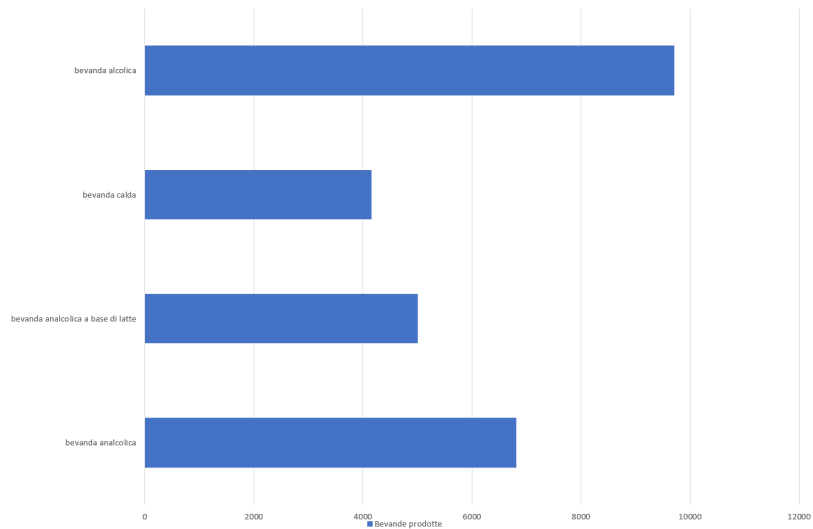


Figure 24

Quindi si ottengono quattro variabili (una per keyword) ognuna delle quali restituisce la somma di tutte le bevande, con quella keyword, prodotte:

```

1  function (keyword)
2  somma = 0
3  for (device in datasources.device)
4      somma += Somma(device.keyword)
5  return somma;

```

5 Caratteristiche "particolari" dei grafici

In questa sezione verranno rielencati tutti i tipi di grafici indicando le loro proprietà "particolari". Le proprietà "particolari" sono proprietà specifiche che non appartengono a tutte le categorie di grafici. I grafici senza caratteristiche "particolari" non sono nella lista.

Le proprietà obbligatorie devono necessariamente essere compilate alla creazione del grafico. Alcune proprietà obbligatorie possono essere compilate automaticamente tramite un valore di default.

5.1 Chart Icon

- icon:
 - descrizione: nome dell'icona da mostrare. Il set dei nomi sarà definito dal generatore.

- tipo: Stringa
- obbligatorio: SI
- valuePosition:
 - descrizione: indica dove viene posizionato il valore rispetto all'icona.
 - tipo: Stringa
 - valori permessi: "left", "right", "top", "bottom"
 - obbligatorio: SI
 - valore di default: "right". Ovvero il valore e l'eventuale additionalText vengono posizionati alla destra dell'icona.
- label:
 - descrizione: testo che descrive il valore mostrato.
 - tipo: Stringa
 - obbligatorio: NO

5.2 Gauge Chart e Gauge Chart con ago

- min:
 - descrizione: valore che corrispondere allo 0%. Se il valore della variabile rappresentata è uguale o minore a min, allora il semicerchio sarà completamente vuoto.
 - tipo: Numero
 - obbligatorio: SI
- max:
 - descrizione: valore che corrispondere al 100%. Se il valore della variabile rappresentata è uguale o maggiore a max, allora il semicerchio sarà completamente pieno.
 - tipo: Numero
 - obbligatorio: SI
- startAngle:
 - descrizione: angolo, rispetto all'asse y, da cui far partire il semicerchio.
 - tipo: Numero
 - valori permessi: tutti i numeri da 0 a -180.
 - obbligatorio: SI
 - valore di default: -180

- **endAngle:**
 - descrizione: angolo, rispetto all'asse y, in cui far terminare il semicerchio.
 - tipo: Numero
 - valori permessi: tutti i numeri da 0 a 180.
 - obbligatorio: SI
 - valore di default: 180
- **label:**
 - descrizione: testo mostrato per indicare cosa rappresenta la percentuale.
 - tipo: Stringa
 - obbligatorio: NO

Per capire meglio le proprietà **startAngle** e **endAngle**, in figura sono riportati alcuni esempi:

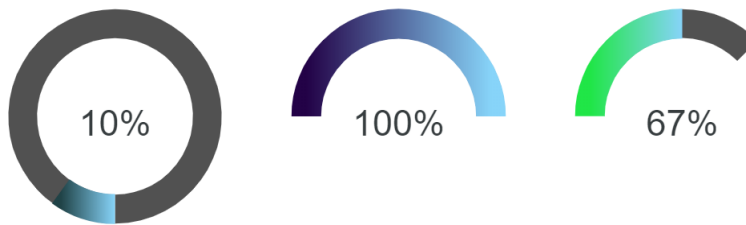


Figure 25: Il primo Gauge ha **startAngle** -180 e **endAngle** 180, il secondo ha **startAngle** -90 e **endAngle** 90, il terzo ha **startAngle** -90 e **endAngle** 45

5.3 Column Chart e Bar Chart

- **mainLabelX:**
 - descrizione: testo che viene posizionato lungo l'asse X
 - tipo: Stringa
 - obbligatorio: NO
- **mainLabelY:**
 - descrizione: testo che viene posizionato lungo l'asse Y.
 - tipo: Stringa
 - obbligatorio: NO

- legendPosition:
 - descrizione: indica dove posizionare la legenda. Se non definita la legenda non viene mostrata.
 - tipo: Stringa
 - valori permessi: "left", "right", "top", "bottom"
 - obbligatorio: NO
 - valore di default: "bottom"

Per esempio, nel grafico in figura 3, è stato posto `mainLabelY = "$ (thousands)"`

5.4 Pie Chart e Donut Chart

- legendPosition:
 - descrizione: indica dove posizionare la legenda. Se non definita la legenda non viene mostrata.
 - tipo: Stringa
 - valori permessi: "left", "right", "top", "bottom"
 - obbligatorio: NO
 - valore di default: "bottom"

5.5 Line Chart

- mainLabelX:
 - descrizione: testo che viene posizionato lungo l'asse X
 - tipo: Stringa
 - obbligatorio: NO
- mainLabelY:
 - descrizione: testo che viene posizionato lungo l'asse Y.
 - tipo: Stringa
 - obbligatorio: NO
- legendPosition:
 - descrizione: indica dove posizionare la legenda. Se non definita la legenda non viene mostrata.
 - tipo: Stringa
 - valori permessi: "left", "right", "top", "bottom"
 - obbligatorio: NO

- valore di default: "bottom"
- lineType:
 - descrizione: valore che indica al generatore come disegnare la stringa
 - tipo: Stringa
 - valori permessi: "basic" per avere una linea semplice, "brush" per ottenere un Brush Line Chart o "point" per avere una linea con i punti che formano la linea in risalto.
 - obbligatorio: NO
 - valore di default: "point"

Il grafico in figura 8 ha lineType uguale a "point". Il Brush Line Chart è mostrato in figura 20.

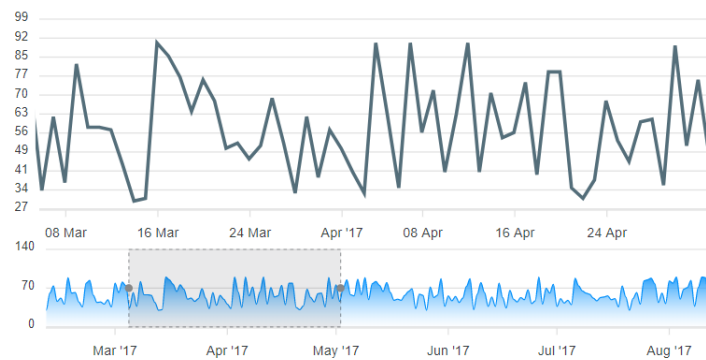


Figure 26: Un Brush Line Chart

5.6 Stacked Area Chart, 100% Stacked Area Chart, Stacked Bar/Column Chart, 100 % Stacked Bar/Column Chart

- mainLabelX:
 - descrizione: testo che viene posizionato lungo l'asse X
 - tipo: Stringa
 - obbligatorio: NO
- mainLabelY:
 - descrizione: testo che viene posizionato lungo l'asse Y.
 - tipo: Stringa

- obbligatorio: NO
- legendPosition:
 - descrizione: indica dove posizionare la legenda. Se non definita la legenda non viene mostrata.
 - tipo: Stringa
 - valori permessi: "left", "right", "top", "bottom"
 - obbligatorio: NO
 - valore di default: "bottom"

5.7 Scatter Plot e Bubble Chart (Forse Non Inclusi)

mainLabelX:

- descrizione: testo che viene posizionato lungo l'asse X
- tipo: Stringa
- obbligatorio: NO

mainLabelY:

- descrizione: testo che viene posizionato lungo l'asse Y.
- tipo: Stringa
- obbligatorio: NO

legendPosition:

- descrizione: indica dove posizionare la legenda. Se non definita la legenda non viene mostrata.
- tipo: Stringa
- valori permessi: "left", "right", "top", "bottom"
- obbligatorio: NO
- valore di default: "bottom"

trendLine:

- descrizione: indica se mostrare la trend line degli sciame di punti.
- tipo: Booleano
- obbligatorio: NO
- valore di default: false

5.8 Histogram

- **classes:**
 - descrizione: lista di oggetti classe. Ogni classe ha tre proprietà; nome, min e max. Il nome è la label che verrà scritta alla base della colonna della classe. Se il nome viene omesso, la label che viene scritta alla base della colonna sarà "min to max". La proprietà min indica il limite sinistro della classe mentre max indica il limite destro.
 - tipo: array di coppie di stringhe
 - valori permessi: qualsiasi numero e le keyword +oo e -oo per indicare -infinito e +infinito, e quindi il bordo sinistro e destro dell'Histogram.
 - obbligatorio: SI
- **mainLabelX:**
 - descrizione: testo che viene posizionato lungo l'asse X
 - tipo: Stringa
 - obbligatorio: NO
- **mainLabelY:**
 - descrizione: testo che viene posizionato lungo l'asse Y.
 - tipo: Stringa
 - obbligatorio: NO
- **legendPosition:**
 - descrizione: indica dove posizionare la legenda. Se non definita la legenda non viene mostrata.
 - tipo: Stringa
 - valori permessi: "left", "right", "top", "bottom"
 - obbligatorio: NO
 - valore di default: "bottom"

Per esempio nella figura 18, la proprietà **categories** viene definita in questo modo: [name: "<0", min: -oo, max: -1,min: 0, max: 14,min: 15, max: 29,min: 30, max: 44, min: 45, max: 59, min: 60, max: 74, name: ">75", min: 75, max: +oo]

6 Fonti

Per la parte dei grafici ho usato le seguenti pagine:

- 365datascience.com/chart-types-and-how-to-select-the-right-one/
- eazybi.com/blog/data-visualization-and-chart-types/
- dbta.com/BigDataQuarterly/Articles/Choosing-the-Right-Chart-Being-Successful-With-Data-Visualization-120616.aspx
- venngage.com/blog/how-to-choose-the-best-charts-for-your-infographic/
- blog.hubspot.com/marketing/types-of-graphs-for-data-visualization
- wikipedia
- docs.microsoft.com/it-it/power-bi/visuals/power-bi-visualization-radial-gauge-charts
- umbertosantucci.it/grafico-a-dispersione/
- datavizcatalogue.com/methods/stacked-area-graph.html
- python-graph-gallery.com/stacked-area-plot/
- <https://www.datapine.com/blog/how-to-choose-the-right-data-visualization-types/>
- **canva.com/graphs/**

Per generare i grafici ho utilizzato:

- live.amcharts.com/
- apexChart React
- www.meta-chart.com
- onlinecharttool.com