10 febbraio 2023
22 febbraio 2023
10 marzo 2023
22 marzo 2023

Data science for Citizens
Analisi di un dataset

1. Cecare il dataset

Abbiamo visto l'altra volta che per trovare un dataset possiamo

- Costruire un sondaggio (google form): costruire un sondaggio è in realtà una cosa più complicata di quello che possiamo pensare: bisogna fare attenzione a come si pongono le domande (non devono essere confuse, bisogna prima pensare bene a cosa si vuole ottenere da una determinata domanda) e anche a come ci si rivolge al pubblico (non si deve usare per esempio ragazza o ragazzo ma ragazz*). Prima di poter diffondere un sondaggio, bisogna accertarsi che sia ben fatto!

L'altra volta abbiamo provato ma per poter essere diffuso dovremmo spenderci molto più tempo, che purtroppo non abbiamo!

1. Cercare il dataset

- Affidarci a dei siti.

Vi lascio qua quelli che abbiamo visto l'altra volta, così nel caso vi servissero in futuro!

https://www.kaggle.com/ (da cui abbiamo preso il nostro dataset)

https://data.world/

https://dataverse.harvard.edu/

https://clinicaltrials.gov/

1. Cercare il dataset

Abbiamo due file:

1) il primo è quello che abbiamo sistemato l'altra volta e di cui abbiamo anche già analizzato il significato delle variabili.

top_influencer_data2.csv

1. Cercare il dataset

- rank: ordinamento in base al numero dei followers
- channel_info: nickname del profilo della persona, nome dell'account di quella persona
- influence score: quanto quell'account è influente sulla base del numero dei followers, sulla base del loro impatto sulle persone (quanto esse riescano a farsi influenzare da quella persona), sulla base di quanto curano l'account. Il massimo può essere 100, serve alle agenzie per capire da chi far sponsorizzare un loro prodotto
- posts: numero dei post che hanno fatto, fino all'analisi. Vengono misurati in K
- followers: numero degli utenti che seguono un account. Vengono misurati in milioni
- average likes: media totale dei likes nei posts. Viene misurata in milioni
- 60 day engagement rate: media tra i likes, condivisioni, commenti, in generale interazioni. Viene misurata in percentuale
  $ER = \frac{like+commenti}{followers} x100$
- new post average like: la media dei likes dei nuovi posts. Viene misurato in milioni
- total likes: likes totali. Vengono misurati in bilioni
- country: paese di provenienza

1. Cercare il dataset

Abbiamo due file:

2) il secondo è un altro file chiamato social media influencers - instagram.csv

1. Cercare il dataset

Abbiamo due file:

2) il secondo è un altro file chiamato

social media influencers - instagram.csv

https://www.kaggle.com/code/booroom/social-media-influencers-basic-data-vis/data

Guardiamolo un attimo.

Cosa ci potrebbe interessare di questo nuovo dataset?

2. Qual è la nostra domanda di ricerca?

Prima di iniziare l'analisi vera e propria dobbiamo capire cosa è una research question e come potremmo definire la nostra.

https://www.scribbr.com/research-process/research-question-examples/

3. Lavoriamo su R: working directory

- Settiamo la working directory.

  Vi ricordate come si fa e cosa si deve fare prima?

3. Lavoriamo su R: working directory

- Settiamo la working directory.

Vi ricordate come si fa e cosa si deve fare prima?

Innanzitutto mettiamo il nostro file in una cartella.

3. Lavoriamo su R: working directory

- Settiamo la working directory.

  Vi ricordate come si fa e cosa si deve fare prima?

Innanzitutto mettiamo il nostro file in una cartella.

Primo modo:

#settiamo la working directory

setwd("D:/SCUOLA/Milano Data Science/Progetto Liceo")

(per mandare un comando cliccare su RUN o CTRL + INVIO)

Per trovare questo path, andiamo nella cartella del file, tasto destro e proprietà.

3. Lavoriamo su R: working directory

- Settiamo la working directory.

  Vi ricordate come si fa e cosa si deve fare prima?

Innanzitutto mettiamo il nostro file in una cartella.

Primo modo:

#settiamo la working directory

setwd("D:/SCUOLA/Milano Data Science/Progetto Liceo")

(per mandare un comando cliccare su RUN o CTRL + INVIO)

Per trovare questo path, andiamo nella cartella del file, tasto destro e proprietà.

Ricordatevi le virgolette!

3. Lavoriamo su R: working directory

- Settiamo la working directory.
  Vi ricordate come si fa e cosa si deve fare prima?

Innanzitutto mettiamo il nostro file in una cartella.

Primo modo:

#settiamo la working directory

setwd("D:/SCUOLA/Milano Data Science/Progetto Liceo")

(per mandare un comando cliccare su RUN o CTRL + INVIO)

Per trovare questo path, andiamo nella cartella del file, tasto destro e proprietà.

Ricordatevi le virgolette!

Secondo modo:

Session → set working directory → choose directory → scegliamo la cartella in cui abbiamo sistemato il file

4. Caricamento del file

- Carichiamo il file: il nostro file è in csv = comma-separated value: i file sono salvati in excel, divisi da una virgola.

4. Caricamento del file

- Carichiamo il file: il nostro file è in csv = comma-separated value: i file sono salvati in excel, divisi da una virgola.
- #carichiamo il file

read.csv("/SCUOLA/Milano Data Science/Progetto Liceo/top_insta_influencers_data2.csv")

4. Caricamento del file

- Carichiamo il file: il nostro file è in csv = comma-separated value: i file sono salvati in excel, divisi da una virgola.
- #carichiamo il file

read.csv("/SCUOLA/Milano Data Science/Progetto Liceo/top_insta_influencers_data2.csv")

È lo stesso path del vostro file, ma alla fine ci aggiungiamo il nome del file.

Ricordatevi le virgolette!

Read csv sta per leggi il file.

4. Caricamento del file

- Carichiamo il file: il nostro file è in csv = comma-separated value: i file sono salvati in excel, divisi da una virgola.
- read.csv(read.csv("/SCUOLA/Milano Data Science/Progetto Liceo/top_insta_influencers_data2.csv")

È lo stesso path del vostro file, ma alla fine ci aggiungiamo il nome del file.

Ricordatevi le virgolette!

Read csv sta per leggi il file.

Dando solo questo comando, cosa pensate manchi?

4. Caricamento del file

- Carichiamo il file: il nostro file è in csv = comma-separated value: i file sono salvati in excel, divisi da una virgola.
- read.csv(read.csv("/SCUOLA/Milano Data Science/Progetto Liceo/top_insta_influencers_data2.csv")

È lo stesso path del vostro file, ma alla fine ci aggiungiamo il nome del file.

Ricordatevi le virgolette!

Read csv sta per leggi il file.

Dando solo questo comando, cosa pensate manchi?

Il file viene sì letto, ma non salvato nell'Environment! E' stato letto ma non assegnato.

4. Caricamento del file

- Carichiamo il file: il nostro file è in csv = comma-separated value: i file sono salvati in excel, divisi da una virgola.
- read.csv(read.csv("/SCUOLA/Milano Data Science/Progetto Liceo/top_insta_influencers_data2.csv")

È lo stesso path del vostro file, ma alla fine ci aggiungiamo il nome del file.

Ricordatevi le virgolette!

Read csv sta per leggi il file.

Dando solo questo comando, cosa pensate manchi?

Il file viene sì letto, ma non salavto nell'Environment! E' stato letto ma non assegnato.

Come si assegna?

4. Caricamento del file

- Carichiamo il file: il nostro file è in csv = comma-separated value: i file sono salvati in excel, divisi da una virgola.

read.csv(read.csv("/SCUOLA/Milano Data Science/Progetto Liceo/top_insta_influencers_data2.csv")

È lo stesso path del vostro file, ma alla fine ci aggiungiamo il nome del file.

Ricordatevi le virgolette!

Read csv sta per leggi il file.

Dando solo questo comando, cosa pensate manchi?

Il file viene sì letto, ma non salavto nell'Environment! E' stato letto ma non assegnato.

- Come si assegna?

dati <- read.csv("/SCUOLA/Milano Data Science/Progetto Liceo/top_insta_influencers_data2.csv")

(p.s. diamo dei nomi brevi, bisognerà richiamarlo molte volte!)

5. Come visualizzare il dataset?

- Vediamo il dataset

1 modo:

#per vedere i dati

View(dati)

2 modo:

Clicchiamo su dati, salvato nell'environment.

Nell'environment ci sono in memoria tutti gli oggetti che salviamo.

6. Vediamo un po' il dataset

- #per avere le prime righe del dataset

head(dati) : ci restituisce le prime righe.

Cosa possiamo vedere in questo modo?

6. Vediamo un po' il dataset

- head(dati) : ci restituisce le prime righe.

Cosa possiamo vedere in questo modo?

I nomi delle colonne e delle righe e come sono salvati i dati.

6. Vediamo un po' il dataset

- head(dati) : ci restituisce le prime righe.

Cosa possiamo vedere in questo modo?

I nomi delle colonne e delle righe e come sono salvati i dati.

- colnames(dati) : vedere i nomi delle colonne
- rownames(dati) : vedere i nomi delle righe

6. Vediamo un po' il dataset

- colnames(dati) : vedere i nomi delle colonne
- rownames(dati) : vedere i nomi delle righe

Sapete come otteniamo con questi due comandi?

Un vettore!

Il vettore

Si può immaginare un array come una sorta di contenitore, le cui caselle sono dette *celle* (o *elementi*) dell'array stesso.

Ciascuna delle celle si comporta come una variabile tradizionale; tutte le celle sono variabili di uno stesso tipo preesistente, detto *tipo base* dell'array. Si parlerà perciò di tipi come "array di interi", "array di stringhe", "array di caratteri" e così via. Quello che si ottiene dichiarandolo è dunque un contenitore omogeneo di valori, variabili o oggetti.

Fonte: https://it.wikipedia.org/wiki/Array

Concentriamoci sul vettore delle righe: cosa notate? Cosa ci dice?

Concentriamoci sul vettore delle righe: cosa notate? Cosa ci dice?

Il numero di elementi del nostro dataset!

Questo è un array unidimensionale, come si trova la dimensione, cioè il numero degli elementi?

Concentriamoci sul vettore delle righe: cosa notate? Cosa ci dice?

Il numero di elementi del nostro dataset!

Questo è un array unidimensionale, come si trova la dimensione, cioè il numero degli elementi?

length(rownames(dati))

Concentriamoci sul vettore delle righe: cosa notate? Cosa ci dice?

Il numero di elementi del nostro dataset!

Questo è un array unidimensionale, come si trova la dimensione, cioè il numero degli elementi?

length(rownames(dati))

Ricordate di assegnare questa variabile, altrimenti R eseguirà il comando ma non salverà nulla.

Per poter restituire il numero che vogliamo, dobbiamo poi richiamare il nome.

Concentriamoci sul vettore delle righe: cosa notate? Cosa ci dice?

Il numero di elementi del nostro dataset!

Questo è un array unidimensionale, come si trova la dimensione, cioè il numero degli elementi?

length(rownames(dati))

Ricordate di assegnare questa variabile, altrimenti R eseguirà il comando ma non salverà nulla.

Per poter restituire il numero che vogliamo, dobbiamo poi richiamare il nome.

n_dati<-length(rownames(dati))

n_dati

7. Caricamento del secondo dataset

```
dati2 <- read.csv("/SCUOLA/Milano Data Science/Progetto Liceo/top_1000_instagrammers.csv")

View(dati2)
```

Osserviamo il dataset: quali variabili ci potrebbero interessare e come potremmo unire i due dataset?

7. Caricamento del secondo dataset

```
dati2 <- read.csv("/SCUOLA/Milano Data Science/Progetto Liceo/top_1000_instagrammers.csv")

View(dati2)
```

Osserviamo il dataset: quali variabili ci potrebbero interessare e come potremmo unire i due dataset? Ci serve una variabile in comune!

8. Uniamo i due dataset

#rinominiamo la variabile nel secondo dataset: così possiamo unirli, abbiamo una variabile comune rispetto cui unire!

dati2$channel_info <- Influencer.insta.name

#facciamo un left join

8. Uniamo i due dataset

#rinominiamo la variabile nel secondo dataset: così possiamo unirli, abbiamo una variabile comune rispetto cui unire!

dati2$channel_info <- Influencer.insta.name

#facciamo un left join: cioè prendiamo tutto il dataset di sinistra e aggiungiamo le informazioni che possiamo ottenere dal dataset di destra.

dataset <- left_join(dati, dati2, by = "channel_info")

write.csv(dataset, "dataset_csv")

str(dataset)

View(dataset)

9. Modifichiamo il nostro nuovo dataset

```
#vogliamo solo i top 100
dataset <- dataset[-c(righe da togliere),]
Dataset
```

Ora vogliamo togliere quali colonne?

Come facciamo a ottenere tutti i nomi delle colonne del nostro nuovo dataset?

9. Modifichiamo il nostro nuovo dataset

```
#vogliamo solo i top 100
dataset <- dataset[-c(righe da togliere),]
dataset
dataset <- dataset[,-c(colonne da togliere)]
library(tidyverse)
dataset <- dataset %>%
  rename(
    id_instagram = channel_info,
    categories = category_1,
    audience_country = Audience.Country)
dataset
```

10. Vediamo meglio la colonna «categories»

```
categories2 <- unique(dataset$categories)
categories2
numberOfCategories <- length(categories)
numberOfCategories
```

Se vediamo il dataset, si può notare che alcuni valori sono mancanti. Dato che non sono troppi e che possiamo avere queste informazioni, scarichiamo il dataset e inseriamole manualmente.

10. Gli NA = not available

- Prima calcoliamo quanti sono questi valori mancanti.

Generalmente si dovrebbero eliminare le righe che contengono questi valori.

```
sum(is.na(dataset))
```

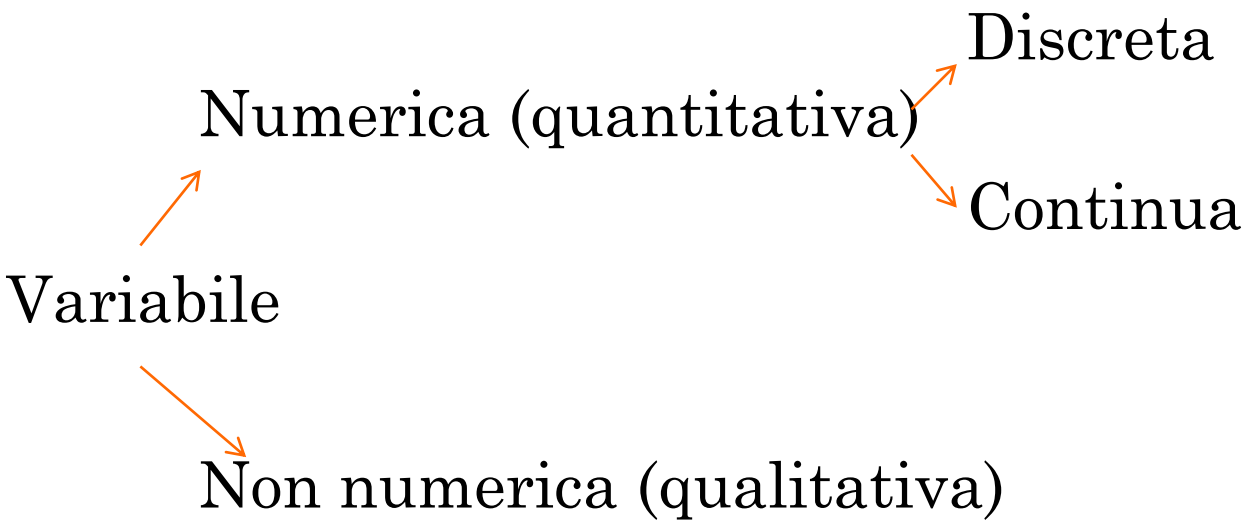
- write.csv(dataset, "D:/SCUOLA/Milano Data Science/Progetto Liceo/Dataset2.csv", row.names=FALSE)

ANALISI UNIVARIATA

ANDIAMO AD ANALIZZARE LE SINGOLE VARIABILI E I VALORI CHE ASSUMONO, INDIPENDENTEMENTE DALLE ALTRE!

11. Analisi delle variabili

Una variabile può essere



Nel nostro dataset?

11. Analisi delle variabili

In R ci sono 6 tipi di dati:

- Logical: boolean data type: TRUE o FALSE
- Numeric: numeri reali con o senza parte decimale
- Integer: numeri reali senza parte decimale
- Complex: numeri complessi
- Character: character o stringhe (‘a’ è un character, ‘apple’ è una stringa, cioè un insieme di caratteri)
- Raw: bytes

11. Analisi delle variabili

- Controlliamo che gli elementi di ciascuna colonna siano corretti, cioè che quelli che devono essere numerici siano numerici e quelli che devono essere testuali siano testuali.

11. Analisi delle variabili

- Controlliamo che gli elementi di ciascuna colonna siano corretti, cioè che quelli che devono essere numerici siano numerici (tipo il numero di likes) e quelli che devono essere testuali siano testuali.
- is.numeric(dati$rank)

Restituisce TRUE o FALSE.

Noi vogliamo che ci restituisca true.

Gli altri controlli

- is.numeric()
- is.character()

Oppure controlliamo con class(). Ci restituisce il tipo di oggetto di quello che c'è nelle parentesi.

Fatelo per ogni variabile che abbiamo!

12. Un po' di analisi descrittiva

Vettori numerici

- min()
- max()
- mean() calcola la media di un vettore di dati (possiamo fare la stessa cosa utilizzando sum e length)

$$MEDIA = \frac{\sum_{i=1}^{n} x_i}{n}$$

- mode <- function(x) {

```
 u <- unique(x)
 tab <- tabulate(match(x, u))
 u[tab == max(tab)]
}
```

12. Un po' di analisi descrittiva

- median() calcola la mediana di un vettore di dati.

E' il valore/modalità (o l'insieme di valori/modalità) assunto dalle unità statistiche che si trovano nel mezzo della distribuzione.

(Solo per quei valori che possono essere ordinati).

- var() calcola la varianza di un vettore di dati, la covarianza tra due vettori, o la matrice di varianze e covarianze di una matrice di dati;

Essa è una misura della variabilità dei valori assunti dalla variabile stessa; nello specifico, la misura di quanto essi si discostino quadraticamente rispettivamente dalla media aritmetica o dal valore atteso.

La covarianza di due variabili statistiche o variabili aleatorie è un valore numerico che fornisce una misura di quanto le due varino assieme.

La matrice varianze-covarianze è una matrice quadrata di dimensione nxm che raccoglie le varianze nella diagonale principale e le covarianze negli elementi esterni alla diagonale principale.

12. Un po' di analisi descrittiva

- cor() calcola la correlazione tra due vettori, o la matrice di correlazione di una matrice di dati;

La *correlazione* indica la tendenza che hanno due variabili (X e Y) a variare insieme, ovvero, a covariare.

- sd() calcola lo scarto quadratico medio (standard error) di un vettore di dati;

La deviazione standard o scarto quadratico medio ci indica le differenze di ogni osservazione rispetto alla media, ci dice quanto può variare un certo valore, un certo fenomeno.

La deviazione standard è l'evoluzione dell'indice di dispersione.

Indice di dispersione solo per poche variabili, altrimenti usiamo la deviazione standard

Il valore starà in media-dispersione e media+dispersione.

12. Un po' di analisi descrittiva

- length() restituisce la lunghezza del vettore (ossia la numerosità campionaria, se si opera su dati campionari);
- sum() calcola la somma degli elementi di un vettore.

12. Un po' di analisi descrittiva

C'è una funzione che ci permette di ottenere molti dei valori che abbiamo calcolato singolarmente prima:

summary()

Ci restituisce i valori minimo e massimo, media, mediana e quartili.

I quartili ripartiscono la popolazione in quattro parti:

1. Primo quartile: ¼ della popolazione
2. Secondo quartile: 2/4 della popolazione
3. Terzo quartile: ¾ della popolazione
4. Quarto quartile: 4/4 della popolazione

I QUARTILI

- Il quartile zero, il primo, il secondo, il terzo e il quarto quartile corrispondono con le prime modalità la cui frequenza cumulata percentuale è almeno 0, 25, 50, 75 e 100 rispettivamente. Cioè, ad esempio, il primo quartile corrisponde con la modalità i-esima se la frequenza cumulata percentuale $P\ i - 1 < 25$ e $P\ i \geq 25$.

- Frequenza assoluta = il numero di volte in cui un dato si ripete
- Frequenza relativa = $\frac{frequenza\ assoluta}{numero\ unità\ statistiche}$
- Frequenza percentuale = frequenza relativa x 100
- Frequenza cumulata assoluta = frequenza assoluta ma ordinate si fa la somma con quella precedente
- Frequenza cumulata relativa = $\frac{frequenza\ cumulata\ assoluta\ iesima}{frequenza\ cumulata\ finale}$
- Frequenza cumulata percentuale = frequenza cumulata relativa x 100

12. Un po' di analisi descrittiva

- library(summarytools)

descr(dataset)

Calcola le statistiche descrittive delle variabili numeriche.

Possiamo darle come parametro anche una sola variabile.

12. Un po' di analisi descrittiva

- library(psych)

describeBy(variabile numerica, variabile categorica)

Otteniamo la descrizione di ogni variabile numerica per ogni tipo di variabile categorica.

FATE L'ANALISI DESCRITTIVA DI OGNI VARIABILE!

13. I BOXPLOT

- I boxplot sono dei grafici utili a descrivere una variabile quantitativa.
- Nel boxplot possiamo vedere cinque valori:
  - Il limite inferiore (non anomalo)
  - Q1
  - Q2 = mediana, il valore centrale della distribuzione
  - Q3
  - Il limite superiore (non anomalo)
- I valori anomali, sopra il limite superiore e sotto il limite inferiore, sono rappresentati dai pallini e sono gli **outliers**.
- Se non ci sono gli outliers:
  - Limite inferiore = valore minimo
  - Limite superiore = valore massimo

13. Boxplot in R

#facciamo un boxplot semplice

boxplot(var_numerica)

- #aggiungiamo informazioni

Boxplot(var_numerica, main = «Titolo grafico», col = «colore della scatola»)

Esempi di colori: red, black, blue…

CREATE I BOXPLOT PER OGNI VARIABILE CHE RITENETE OPPORTUNA.

14. Gli Outliers

```
outliers_influence_score <-
boxplot.stats(dataset$influence_score)
```

Stampiano questo comando.

Che cosa otteniamo? Come facciamo a vedere i valori degli outliers?

14. Gli Outliers

```
outliers_influence_score <-
boxplot.stats(dataset$influence_score)
```

Stampiano questo comanda.

Che cosa otteniamo? Come facciamo a vedere i valori degli outliers?

```
outliers_influence_score <-
boxplot.stats(dataset$influence_score)$out
```

14. Gli Outliers

Ora, ottenuti i valori degli outliers, che cosa vorremmo vedere?

14. Gli Outliers

Ora, ottenuti i valori degli outliers, che cosa vorremmo vedere?

```
persone <- dataset[dataset$influence_score %in% outliers_influence_score, ]$id_instagram
persone
```

15. L'ISTOGRAMMA

- Sapete cosa è un istogramma?
- L'istogramma si usa per visualizzare la distribuzione di una variabile numerica continua.

15. L'ISTOGRAMMA

- Sapete cosa è un istogramma?
- L'istogramma si usa per visualizzare la distribuzione di una variabile numerica continua.

- hist(variabile,
  col = «»,
  breaks =seq(from = valore iniziale, to = valore finale, by = step),
  xlab = «nome asse x»,
  ylab=«nome asse y»,
  main=«Titolo del grafico»)

15. L'ISTOGRAMMA CON GGPLOT

- Installiamo la libreria

```
library(ggplot2)

ggplot(nome dataset, aes(x = nome variabile continua))+
geom_histogram(bindwidth = , colour = «colore del bordo», fill = «colore di riempimento»)+
labs(title = «Titolo del grafico»,
        x = «nome asse x»,
        y = «nome asse y»)
```

Il comando ggplot è molto generale, serve a fare dei grafici. Con il + aggiungiamo il tipo di grafico da fare.

15. L'ISTOGRAMMA CON GGPLOT

- Con ggplot possiamo anche fare l'istogramma per gruppi

ggplot(nome dataset, aes(x = nome variabile continua, fill = nome variabile qualitativa))+

geom_histogram(bindwidth = , colour = «colore del bordo», fill = «colore di riempimento»)+

labs(title = «Titolo del grafico»,

x = «nome asse x»,

y = «nome asse y»)

16. La variabile country

- Per la variabile COUNTRY introduciamo il grafico a barre.
- Il grafico a barre si usa per variabili categoriche: assomiglia all'istogramma e si hanno tanti segmenti (barre) quante sono le modalità del carattere, aventi lunghezza proporzionale alla frequenza assoluta o relativa.
- Qual è la differenza con l'istogramma?
  - Istogramma → variabili continue
  - Grafico a barre → variabili discrete

16. La variabile country

- Plottiamo il grafico a barre usando ggplot
- ggplot(dataset, aes(x = country)) +
  geom_bar()
- Aggiungiamo la quantità per ogni barra
- ggplot(dataset, aes(x= country)) +
  geom_bar()+
  geom_text(stat ='count', aes(label=..count..), vjust=-0.3)

16. La variabile country

OSSERVAZIONI

- In quale country si hanno il maggior numero di influencers?
- Qual è la seconda nazione che ha il maggior numero di influencers?
- Chi manca tra tutte le nazioni?

- https://www.digitalmarketingcommunity.com/indicators/instagram-active-users-penetrations-2018/

16. La variabile audience_country

Facciamo la stessa cosa con la variabile audience_country!

17. La variabile categories

- Facciamo la stessa cosa con la variabile categories per capire quale attività fanno gli influencers.
- Quali sono i principale lavori?
- Cosa ci dice questo?

ANALISI BIVARIATA

STUDIO COINGIUNTO DI DUE VARIABILI O MUTABILI STATISTICHE, PER VALUTARE L'ESISTENZA DI UN EVENTUALE LEGAME FRA LE STESSE!

Tabella di contingenza e tabella di correlazione

Si parla di:

- **tabella di contingenza** quando i fenomeni considerati sono entrambi qualitativi oppure uno qualitativo e l'altro quantitativo
- **tabella di correlazione** quando X e Y sono entrambi quantitativi.

MATRICE DI CORRELAZIONE

- Una matrice è una tabella e in ogni cella c'è un valore.
- La matrice di correlazione è quadrata, cioè ha lo stesso numero di righe.
- Nella diagonale c'è il grado di correlazione di una variabile con se stessa: è totale! Il valore sarà sempre 1.
- Sopra e sotto la diagonale ci saranno i gradi di correlazione tra le varie variabili.

Ovviamente la variabile A e la variabile B sono correlate nello stesso modo in cui sono correlate la variabile B e la variabile A: l'ordine non interessa => la matrice sarà simmetrica rispetto alla diagonale!

I COEFFICIENTI DI CORRELAZIONE

- I coefficienti di correlazione descrivono la relazione tra le variabili.
  - Indice di Pearson → si può usare solo se le due variabili hanno una distribuzione normale, se la relazione è lineare e se non ci sono outliers.
  - Indice di Spearman
  - Indice di Kendall

  Non richiedono né la normalità distributiva delle variabili, né che la relazione sia lineare.

  Prima di costruire una matrice di correlazione è comunque sempre buona norma valutare la relazione tra le coppie di variabili attraverso la costruzione di diagrammi di dispersione.

Diagrammi a dispersione - scatterplot

- Facciamo gli scatterplot per vedere se c'è una relazione tra le variabili.

plot(prima variabile, seconda variabile)

Matrice di correlazione

```
library(Hmisc)
library(corrplot)
dataset.rcorr = rcorr(as.matrix(dataset[,c(3:9)]))
dataset.rcorr
dataset.cor2 = cor(dataset[,c(3:9)], method = c("spearman"))
dataset.cor2
```