Data Warehousing Seven_App

Salvatore Ferrandino, Gennaro Francesco Landi, Lorenzo Gravina, Matteo Cavallaro

2025-05-13

Abbiamo deciso di realizzare il nostro piccolo esempio di data warehousing servendoci del linguaggio di programmazione R, poiché esso è finalizzato al data analysis; inoltre, presenta una sintassi molto semplice e sono nativamente presenti strutture dati molto utili, in particolare il data.frame, di cui ci serviremo in questa analisi. Inoltre, mediante l'implementazione con Markdown tramite RStudio, rappresenta un'ottima soluzione anche nell'ottica di produzione di report.

Innanzitutto, importiamo nella sessione di R il file .csv esportato da MySQL Workbench

```
fact_table <- read.csv("C:/Users/rtr-f/Desktop/da uppare/fact_table.csv", sep = ";")
```

Tramite R, possiamo agire sul data-frame ottenuto a partire dal .csv per andare a sintetizzare delle informazioni che possono essere utili al management per prendere decisioni. In particolare, ci interessa sapere il periodo del giorno in cui vengono pubblicati più post; in tal caso, possiamo, tramite un ciclo for, andare a costruire una nuova variabile, ottenuta in questo modo:

```
library(lubridate) # necessario installare lubridate, se non è già installato
```

```
##
## Caricamento pacchetto: 'lubridate'
```

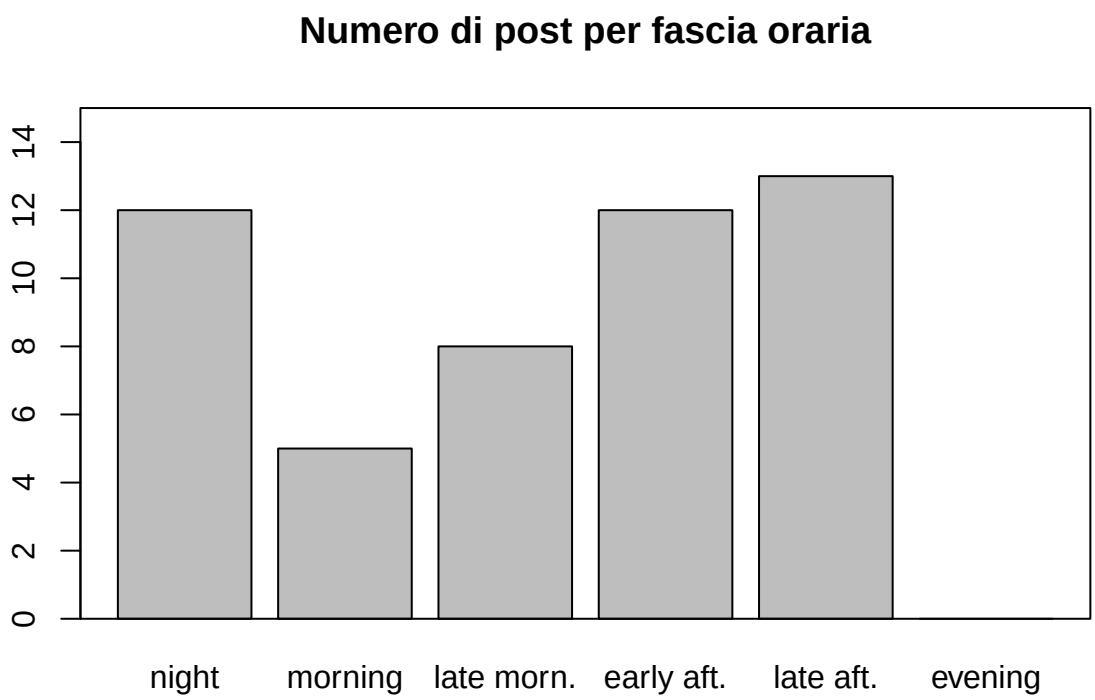
```
## I seguenti oggetti sono mascherati da 'package:base':
##
##     date, intersect, setdiff, union
```

```
fact_table$time_of_day <- rep(0, nrow(fact_table))
fact_table$Data_ora <- ymd_hms(fact_table$Data_ora)

for (ii in 1:nrow(fact_table)) {
  if (0 <= hour(fact_table$Data_ora[ii]) & hour(fact_table$Data_ora[ii]) <= 4) {
    fact_table$time_of_day[ii] <- "night"
  }else{
    if (5 <= hour(fact_table$Data_ora[ii]) & hour(fact_table$Data_ora[ii]) <= 8) {
      fact_table$time_of_day[ii] <- "early morning"
    }else{
      if (9 <= hour(fact_table$Data_ora[ii]) & hour(fact_table$Data_ora[ii]) <= 12) {
        fact_table$time_of_day[ii] <- "late morning"
      }else{
        if (13 <= hour(fact_table$Data_ora[ii]) & hour(fact_table$Data_ora[ii]) <= 16) {
          fact_table$time_of_day[ii] <- "early afternoon"
        }else{
```

```
          if (17 <= hour(fact_table$Data_ora[ii]) & hour(fact_table$Data_ora[ii]) <= 19) {
            fact_table$time_of_day[ii] <- "late afternoon"
          }else{
            fact_table$time_of_day[ii] <- "evening"
          }
        }
      }
    }
  }
}

fact_table$time_of_day <- as.factor(fact_table$time_of_day)
levels(fact_table$time_of_day) <- c("night", "early morning", "late morning",
                                    "early afternoon", "late afternoon",
                                    "evening")
```

Possiamo poi costruire un barplot che conti il numero di post per fascia oraria.

```
barplot(height = table(fact_table$time_of_day), main = "Numero di post per fascia oraria",
        names.arg = c("night", "morning", "late morn.", "early aft.", "late aft.",
                      "evening"), ylim = c(0,15))
box()
```



Ora che sappiamo come la maggior parte dei post venga pubblicata tra la notte e il pomeriggio, possiamo ad esempio pensare a tariffe personalizzate per le inserzioni pubblicitarie o a un miglioramento dell'algoritmo

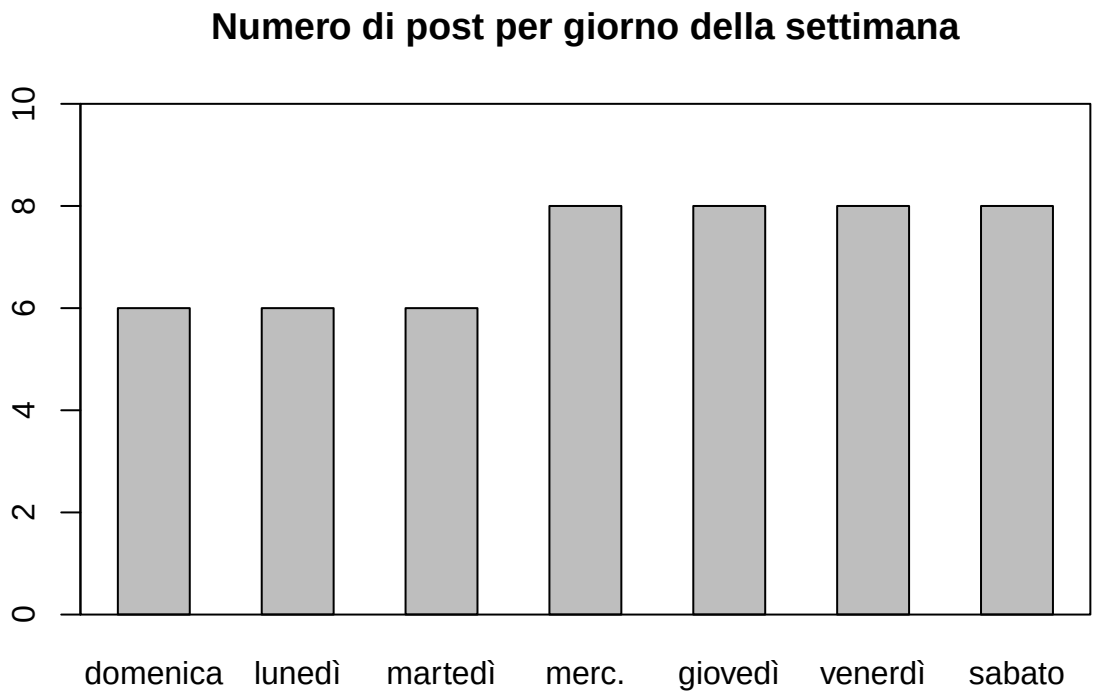
di raccomandazione dei contenuti, magari studiando la distribuzione condizionata dei gruppi di tag per fasce orarie.

Tramite lubridate, possiamo anche ricavare i giorni della settimana a partire dalla variabile Data_ora.

```
fact_table$weekday <- wday(fact_table$Data_ora)
fact_table$weekday <- as.factor(fact_table$weekday)
levels(fact_table$weekday) <- c("domenica", "lunedì", "martedì", "mercoledì",
                                "giovedì", "venerdì", "sabato")
```

Di conseguenza, possiamo realizzare un barplot che conti il numero di post realizzati per ogni giorno della settimana.

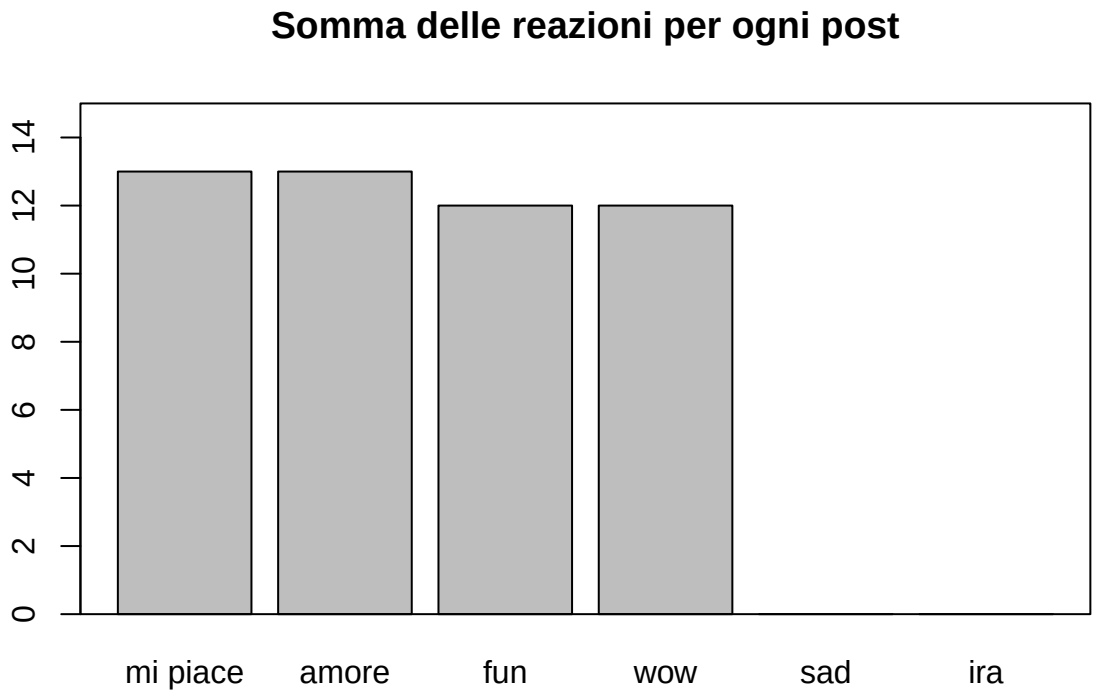
```
barplot(height = table(fact_table$weekday),
        main = "Numero di post per giorno della settimana",
        names.arg = c("domenica", "lunedì", "martedì", "merc.",
                      "giovedì", "venerdì", "sabato"),
        space = 1, ylim = c(0,10))
box()
```



Da questa prima analisi, possiamo vedere come i giorni che vanno dal mercoledì al sabato siano quelli in cui si assiste a una maggiore produzione di contenuti. Questo magari potrebbe esserci utile alle stesse finalità dell'analisi precedente.

Possiamo anche vedere quali sono le reazioni più utilizzate, allo scopo di comprendere lo stato d'animo generale della nostra utenza.

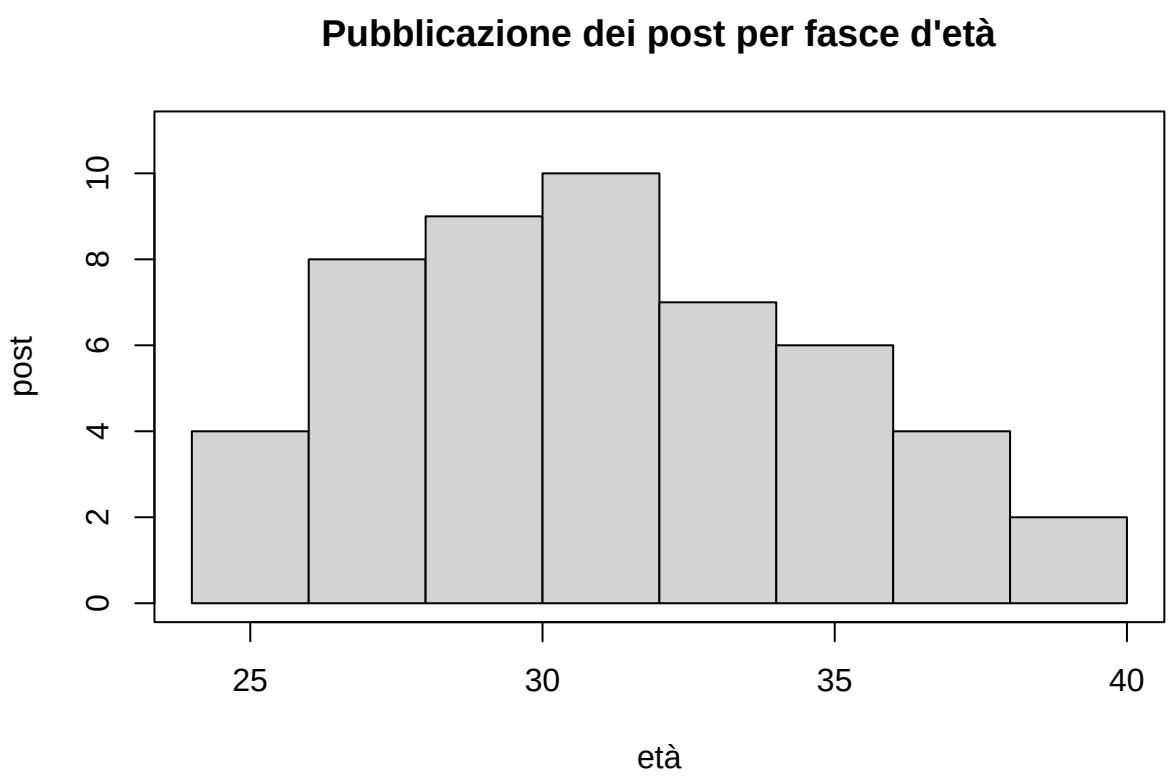
```r
barplot(height = c(sum(fact_table$number_of_mi_piace),
                   sum(fact_table$number_of_amore),
                   sum(fact_table$number_of_divertente),
                   sum(fact_table$number_of_wow),
                   sum(fact_table$number_of_triste),
                   sum(fact_table$number_of_ira)),
        main = "Somma delle reazioni per ogni post",
        names.arg = c("mi piace", "amore", "fun", "wow", "sad", "ira"),
        ylim = c(0,15))
box()
```



Da questo boxplot, possiamo vedere come mi piace, amore, fun e wow siano le reazioni più utilizzate, mentre sad e ira sono molto poco utilizzate. Per cui, potrebbe essere utile spingere sulla piattaforma contenuti che abbiano un tono allegro o positivo.

In ultima analisi, possiamo andare a studiare la pubblicazione dei post in base alle fasce d'età

```r
hist(fact_table$eta, breaks = "FD", main = "Pubblicazione dei post per fasce d'età", xlab = "età", ylab
box()
```



Da questo istogramma possiamo vedere come la maggior parte dei post sia pubblicata da un'utenza che ha tra i 28 e i 32 anni circa, con un trend decrescente per le due code. Questo tipo di analisi, magari legata a una successiva analisi specifica sui trend, potrebbe guidare meglio il management a determinare meglio gli algoritmi di raccomandazione.

Questo report vuole essere un piccolo esempio solo di alcune delle informazioni che si potrebbero ricavare, mediante strumenti elementari di analisi esplorativa dei dati, a partire da una struttura di data warehousing di un social media.