

Università degli Studi di Torino

Facoltà di Economia e Management

Corso di laurea in Amministrazione Aziendale



**Relazione operativa**

del corso Big Data and Analytics

**Il Data Frame in R: struttura e applicazioni**

Massimo Varesio

Matricola [REDACTED]

Anno Accademico 2020/2021

# Abstract

La trattazione seguente ha come obiettivo quello di effettuare un'analisi panoramica dell'ambiente R, in particolare riguardo ad uno degli oggetti principali: il *dataframe*. Come vedremo, partendo da un set di dati correttamente composto, è possibile condurre un'analisi esplorativa finalizzata ad evidenziare quali sono le caratteristiche peculiari di una popolazione presa in esame.

La presente relazione, inizierà con una breve introduzione riguardante la descrizione del *dataframe* e le principali proprietà, per poi concludersi con lo sviluppo di un caso di studio pratico, applicato su un dataset relativo allo stato dell'economia in Taiwan nell'anno 2005.

Ho scelto di svolgere una relazione di tipo operativo in quanto, benché nuovo in ambiente R, nutro da sempre un forte interesse per le scienze informatiche. Inoltre, svolgendo come professione quella del Front-end Developer, ho trovato un buon riscontro nello sviluppo del tema dal punto di vista pratico, considerando anche la familiarità nel reperire un'ampia documentazione online, semplicemente navigando piattaforme web utilizzate nel lavoro quotidiano, come ad esempio Github o AWS.

# Indice

- 1 Il dataframe**
- 2 Data processing con R**
- 3 Applicazione di funzioni**
- 4 Alcuni esercizi sul dataframe**
- 5 L'analisi esplorativa dei dati - Il caso di studio**
  - 5.2 Introduzione**
  - 5.3 Il dataset utilizzato**
  - 5.4 Importazione dei dati**
  - 5.5 Cleaning**
  - 5.6 Analisi delle variabili**
  - 5.7 Boxplot e conclusioni**
- 6 Bibliografia**

# 1. Il dataframe

Il *dataframe* costituisce l'oggetto più importante di tutto l'ambiente R, almeno per quanto riguarda la gestione e analisi dei dati. Il *dataframe* rappresenta una matrice di dati, in cui ad ogni riga corrisponde una osservazione e ad ogni colonna una variabile. In termini più formali, un *dataframe* può essere definito come una “*lista di classe data.frame*”.<sup>1</sup>

Nell'ambiente viene quindi trattato come una lista, in cui ogni elemento rappresenta una variabile statistica. La funzione `str()` restituisce la struttura delle variabili, mentre `names()` i rispettivi nomi. Tra le diverse opzioni disponibili, è possibile costruire un *dataframe* direttamente con la funzione `data.frame()`.<sup>2</sup>

## 2. Data processing con R

Come abbiamo detto, il *dataframe* può essere considerato il tipo principale di struttura presente in ambiente R. Si tratta di una tabella in cui ogni colonna, contraddistinta da un nome, contiene valori di una specifica variabile. Nella pratica, un `data.frame` viene caricato mediante l'importazione di dati provenienti da una sorgente esterna. In alternativa, quando la mole di dati lo permette, è possibile creare il *dataframe* tramite righe di codice, ad esempio mediante una serie di vettori, come vediamo nell'esempio riportato.

```
> persone <- data.frame(nomi=c("Franco", "Eugenio", "Andrea", "Marco",
+ "Lorenzo", "Marco"),
+ cognomi=c("Rossi", "Bianchi", "Neri", "Rossi", "Gialli", "Bianchi"),
+ eta=c(45, 61, 18, 29, 33, 54),
+ citta=c("Roma", "Milano", "Napoli", "Roma", "Torino", "Verona"))
> persone
  nomi cognomi eta citta
1 Franco Rossi 45 Roma
2 Eugenio Bianchi 61 Milano
3 Andrea Neri 18 Napoli
4 Marco Rossi 29 Roma
5 Lorenzo Gialli 33 Torino
6 Marco Bianchi 54 Verona
```

Una volta acquisito possesso dei dati, esistono diversi modi per prelevarli.

<sup>1</sup> <https://cran.r-project.org/> - “Una guida all'utilizzo dell'ambiente statistico R” by Angelo M. Mineo, an introductory guide, based mainly on “An introduction to R”.

<sup>2</sup> Cfr. <https://cran.r-project.org/> - “Il linguaggio R: concetti introduttivi ed esempi” (II edizione) by Vito M. R. Muggeo and Giancarlo Ferrara

Possiamo, ad esempio, effettuare un prelievo di dati dalle righe o dalle colonne.

## ■ Righe o singoli valori

```
> persone[2,] #come se fosse un record di una classica tabella  
    nomi cognomi eta citta  
2 Eugenio Bianchi 61 Milano  
> persone[2,"eta"] #praticamente una cella della tabella  
[1] 61
```

## ■ Colonne

```
> persone["eta"]  
eta  
1 45  
2 61  
3 18  
4 29  
5 33  
6 54  
> persone[3]  
eta  
1 45  
2 61  
3 18  
4 29  
5 33  
6 54  
> persone$eta  
[1] 45 61 18 29 33 54  
> persone[["eta"]]  
[1] 45 61 18 29 33 54
```

Da notare che, nel caso delle colonne, i primi due esempi di prelevamento del campo eta restituiscono un *dataframe*, mentre il terzo e quarto restituiscono un *vettore*.

In R è altresì possibile effettuare delle sintesi applicando dei filtri sui dati.

```
> persone[persone$eta>29,]      #Selezioniamo tutti i soggetti con più di  
    nomi cognomi eta citta  
29 anni.  
1 Franco Rossi 45 Roma  
2 Eugenio Bianchi 61 Milano  
5 Lorenzo Gialli 33 Torino  
6 Marco Bianchi 54 Verona
```

Si può ricorrere alla funzione `subset()` per svolgere selezioni più precise, come la seguente:

```
> subset(persone, eta>20 & cognomi=="Bianchi")
   nomi cognomi eta citta
2 Eugenio Bianchi 61 Milano
6 Marco Bianchi 54 Verona
```

Fino ad ora si sono osservati esempi poco estesi, ma solitamente la mole di dati che si deve processare è molto più grande. Per questo motivo, esistono in R alcune funzioni che hanno lo scopo di "riassumere" brevemente i dati e creare uno schema visivo per una lettura più semplice e immediata. Fra queste, la funzione denominata `summary()` ci viene spesso in aiuto.

```
> summary(persone)
   nomi      cognomi      eta      citta
Andrea :1    Bianchi:2    Min.   :18.00  Milano:1
Eugenio:1   Gialli :1    1st Qu.:30.00  Napoli:1
Franco  :1    Neri   :1    Median  :39.00  Roma   :2
Lorenzo:1   Rossi  :2    Mean    :40.00  Torino:1
Marco   :2                    3rd Qu.:51.75  Verona:1
                           Max.    :61.00
```

Questa funzione `summary()` esplora la tabella per colonne e offre un conteggio della distribuzione dei valori per le variabili città, nomi e cognomi, mentre sull'età, essendo numeri interi, svolge in aggiunta qualche operazione statistica basilare in merito agli indici di posizione, come il calcolo della media, il massimo, il minimo e i quartili. Un'altra funzione che ci permette di approfondire la conoscenza del `dataframe` è la `str()`, che ha lo scopo di indagare la struttura del dataset presentato.

```
> str(persone)
'data.frame': 6 obs. of 4 variables:
 $ nomi   : Factor w/ 5 levels "Andrea","Eugenio",...: 3 2 1 5 4 5
 $ cognomi: Factor w/ 4 levels "Bianchi","Gialli",...: 4 1 3 4 2 1
 $ eta    : num 45 61 18 29 33 54
 $ citta  : Factor w/ 5 levels "Milano","Napoli",...: 3 1 2 3 4 5
```

In questo risultato emerge un elemento di tipo *Factor*, che in R rappresenta ciò che in Statistica viene chiamata variabile categoriale. Il fattore, solitamente, è utile per raggruppare le unità sottoposte a osservazione<sup>3</sup>.

<sup>3</sup> Cfr. <https://www.html.it/guide/guida-r/>

### 3. Applicazione di funzioni

In R è presente un'ampia varietà di funzioni, comunemente usate nell'ambito delle scienze statistiche. Inoltre, come ogni linguaggio di programmazione, R consente all'utilizzatore di creare funzioni secondo le proprie necessità, e di riutilizzarle semplicemente richiamando la funzione stessa.

Il fine di questa trattazione non è quello di addentrarci nei meandri della disciplina statistica e delle relative funzioni di R. Ci limitiamo quindi a fare alcuni esempi di funzioni basilari che posso essere applicate ad un *dataframe*.

Ad esempio, supponiamo di voler conoscere l'età media, massima o minima delle persone presenti nel data . f rame usando in precedenza<sup>4</sup>.

```
> mean(persone$eta)
[1] 40
> min(persone$eta)
[1] 18
> max(persone$eta)
[1] 61
```

Non adatta a questo esempio, ma occorre menzionare anche la funzione `sum()`, che restituisce come risultato la somma di una serie di valori.

Esistono poi le funzioni aggregative, che possono risultare particolarmente utili in caso di raggruppamento dei dati. Uno dei modi per svolgere ciò in R consiste nell'impiego della funzione `tapply()`, tramite la quale è possibile manovrare e integrare l'utilizzo di altre funzioni, come quelle appena citate.

Nell'esempio riportato effettuiamo il calcolo dell'età media per città di appartenenza:

```
> X<-tapply(persone$eta, persone$citta,max)
> X
Milano Napoli Roma Torino Verona
  61      18     45     33     54
```

`#la funzione è indicata al terzo parametro`

Seguiranno alcuni brevi esempi relativi all'utilizzo dei *dataframe* in R, per poi concludere la trattazione con un esercizio applicativo che mostrerà lo svolgimento a livello basilare di un' analisi esplorativa dei dati, condotta su un *dataframe* di medie proporzioni.

<sup>4</sup> Cfr. <https://www.html.it/pag/65133/data-processing-con-r/>

## 4. Alcuni esercizi sul dataframe

Al fine di dare un ulteriore riscontro pratico dell'utilizzo del `data.frame` in ambiente R, si presentano di seguito alcuni degli esercizi contenuti nella raccolta sul sito web [r-exercises.com<sup>5</sup>](https://www.r-exercises.com/start-here-to-learn-r). Per lo svolgimento degli esercizi si è utilizzata la R console<sup>6</sup>.

- **Esercizio 1:** creare un dataframe secondo le specifiche della richiesta, e successivamente invertire la variabile Sesso per tutti i soggetti.

```
> Nome <- c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas",  
"Caroline")  
Eta <- c(25, 31, 23, 52, 76, 49, 26)  
Alt <- c(177, 163, 190, 179, 163, 183, 164)  
Peso <- c(57, 69, 83, 75, 70, 83, 53)  
Sesso <- as.factor(c("F", "F", "M", "M", "F", "M", "F"))  
df <- data.frame (row.names = Nome, Eta, Alt, Peso, Sesso)  
df
```

	Eta	Alt	Peso	Sesso
Alex	25	177	57	F
Lilly	31	163	69	F
Mark	23	190	83	M
Oliver	52	179	75	M
Martha	76	163	70	F
Lucas	49	183	83	M
Caroline	26	164	53	F

```
> Nome <- c("Alex", "Lilly", "Mark", "Oliver", "Martha", "Lucas",  
"Caroline")  
Eta <- c(25, 31, 23, 52, 76, 49, 26)  
Alt <- c(177, 163, 190, 179, 163, 183, 164)  
Peso <- c(57, 69, 83, 75, 70, 83, 53)  
Sesso <- as.factor(c("F", "F", "M", "M", "F", "M", "F"))  
df <- data.frame (row.names = Nome, Eta, Alt, Peso, Sesso)  
levels(df$Sesso) <- c("M", "F")  
df
```

	Eta	Alt	Peso	Sesso
Alex	25	177	57	M
Lilly	31	163	69	M
Mark	23	190	83	F
Oliver	52	179	75	F
Martha	76	163	70	M
Lucas	49	183	83	F
Caroline	26	164	53	M

<sup>5</sup> <https://www.r-exercises.com/start-here-to-learn-r>

<sup>6</sup> <https://cran.r-project.org/mirrors.html>

- **Esercizio 2:** creare un unico dataframe combinando altri dataframe mediante la funzione `merge()`. Questa funzione simula la funzionalità di unione di database in linguaggio SQL.

```

> edifici <- data.frame(locazione=c(1, 2, 3),
  nome=c("edificio1", "edificio2", "edificio3"))      #creiamo i dataframe da
                                                       unire

> dati <- data.frame(locazione=c(1,2,3,2,3,1),
  efficienza=c(51,64,70,71,80,58))

> statoEdifici <- merge (edifici, dati,
  by = "locazione" )                                     #usiamo la funzione
                                                       merge per unire in un
                                                       singolo dataframe

> statoEdifici

```

	locazione	nome	efficienza
1	1	edificio1	51
2	1	edificio1	58
3	2	edificio2	64
4	2	edificio2	71
5	3	edificio3	70
6	3	edificio3	80

## 5. L'analisi esplorativa dei dati

**Caso di studio:** analisi sui pagamenti inadempienti in Taiwan nel 2005.

### 5.2 Introduzione

L'economia taiwanese ha registrato una crescita enorme durante gli anni '90, quasi raddoppiando il suo valore insieme agli altri paesi conosciuti come le Tigri Asiatiche. Il settore finanziario del paese è stato fortemente coinvolto nella crescita del settore immobiliare durante questo periodo. Tuttavia, all'inizio degli anni 2000, questa crescita è rallentata e le banche di Taiwan si sono rivolte ai prestiti al consumo per continuare l'espansione. Di conseguenza, i requisiti di credito sono stati allentati, e i consumatori sono stati incoraggiati a spendere prendendo in prestito capitale. Analizzeremo i dati sui titolari di carte di credito taiwanesi a partire dalla metà del 2005, quando la voragine del debito ha raggiunto il suo picco<sup>7</sup>.

<sup>7</sup> <https://github.com/Avani10/Taiwan-Credit-Default-Analysis-in-R>

## 5.3 Il dataset utilizzato

Il set di dati sul quale verrà condotto il lavoro contiene dati demografici e di pagamento di 30000 clienti di carte di credito a Taiwan, dall'Aprile 2005 al Settembre 2005. Il dataset proviene dall'Istituto *UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science*<sup>8</sup>, e i dati sono stati prelevati dalla raccolta presente sulla piattaforma web per sviluppatori Github<sup>9</sup>. L'analisi verrà condotta mediante l'applicativo RStudio<sup>10</sup>.

**Numero di istanze:** 30000

**Descrizione delle variabili:** questa ricerca include 25 variabili, di cui una variabile booleana `default_payment_next_month` a cui corrispondono valori Si=1, No=2.

`ID`: ID di ogni cliente

`LIMIT_BAL`: importo del credito concesso in dollari NT (include credito individuale e familiare / supplementare)

`SEX`: sesso (1 = maschio, 2 = femmina)

`EDUCATION`: (1 = scuola di specializzazione, 2 = università, 3 = scuola superiore, 4 = altri)

`MARRIAGE`: Stato civile (1 = sposato, 2 = single, 3 = altri)

`AGE`: Età in anni

`PAY_0`: stato del rimborso a settembre 2005 (-1 = paga debitamente, 1 = ritardo di pagamento per un mese, 2 = ritardo di pagamento per due mesi, ... 8 = ritardo di pagamento per otto mesi, 9 = ritardo di pagamento per nove mesi e oltre)

`PAY_2`: stato del rimborso nell'agosto 2005 (scala come sopra)

`PAY_3`: stato del rimborso nel luglio 2005 (scala come sopra)

`PAY_4`: stato del rimborso nel giugno 2005 (scala come sopra)

`PAY_5`: stato del rimborso a maggio 2005 (scala come sopra)

`PAY_6`: stato del rimborso nell'aprile 2005 (scala come sopra)

`BILL_AMT1`: importo dell'estratto conto nel settembre 2005 (dollaro NT)

`BILL_AMT2`: importo dell'estratto conto dell'agosto 2005 (dollaro NT)

`BILL_AMT3`: importo dell'estratto conto nel luglio 2005 (dollaro NT)

`BILL_AMT4`: importo dell'estratto conto nel giugno 2005 (dollaro NT)

`BILL_AMT5`: importo dell'estratto conto di maggio 2005 (dollaro NT)

`BILL_AMT6`: importo dell'estratto conto nell'aprile 2005 (dollaro NT)

`PAY_AMT1`: importo del pagamento precedente nel settembre 2005 (dollaro NT)

`PAY_AMT2`: importo del pagamento precedente nell'agosto 2005 (dollaro NT)

`PAY_AMT3`: importo del pagamento precedente nel luglio 2005 (dollaro NT)

`PAY_AMT4`: importo del pagamento precedente nel giugno 2005 (dollaro NT)

`PAY_AMT5`: importo del pagamento precedente a maggio 2005 (dollaro NT)

`PAY_AMT6`: importo del pagamento precedente nell'aprile 2005 (dollaro NT)

`default.payment.next.month`: pagamento predefinito nel giugno 2005 (1 = sì, 0 = no)

<sup>8</sup> <https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>

<sup>9</sup> [https://github.com/Avani10/Taiwan-Credit-Default-Analysis-in-R/blob/master/Data/Taiwan\\_credit\\_default\\_data\\_final\\_v1.csv](https://github.com/Avani10/Taiwan-Credit-Default-Analysis-in-R/blob/master/Data/Taiwan_credit_default_data_final_v1.csv)

<sup>10</sup> <https://rstudio.com/>

Al fine rendere più agevole l'elaborazione di questi dati nel codice, ho deciso di modificare il file .csv dal quale è stato prelevato il dataset, sostituendo i nomi delle variabili come riportato di seguito.

**ID:** ID di ogni cliente

**X1** <- LIMIT\_BAL: importo del credito concesso in dollari NT (include credito individuale e familiare / supplementare)  
**X2** <- SEX: sesso (1 = maschio, 2 = femmina)  
**X3** <- EDUCATION: (1 = scuola di specializzazione, 2 = università, 3 = scuola superiore, 4 = altri)  
**X4** <- MARRIAGE: Stato civile (1 = sposato, 2 = single, 3 = altri)  
**X5** <- AGE: Età in anni  
**X6** <- PAY\_0: stato del rimborso a settembre 2005 (-1 = paga debitamente, 1 = ritardo di pagamento per un mese, 2 = ritardo di pagamento per due mesi, ... 8 = ritardo di pagamento per otto mesi, 9 = ritardo di pagamento per nove mesi e oltre)  
**X7** <- PAY\_2: stato del rimborso nell'agosto 2005 (scala come sopra)  
**X8** <- PAY\_3: stato del rimborso nel luglio 2005 (scala come sopra)  
**X9** <- PAY\_4: stato del rimborso nel giugno 2005 (scala come sopra)  
**X10** <- PAY\_5: stato del rimborso a maggio 2005 (scala come sopra)  
**X11** <- PAY\_6: stato del rimborso nell'aprile 2005 (scala come sopra)  
**X12** <- BILL\_AMT1: importo dell'estratto conto nel settembre 2005 (dollaro NT)  
**X13** <- BILL\_AMT2: importo dell'estratto conto dell'agosto 2005 (dollaro NT)  
**X14** <- BILL\_AMT3: importo dell'estratto conto nel luglio 2005 (dollaro NT)  
**X15** <- BILL\_AMT4: importo dell'estratto conto nel giugno 2005 (dollaro NT)  
**X16** <- BILL\_AMT5: importo dell'estratto conto di maggio 2005 (dollaro NT)  
**X17** <- BILL\_AMT6: importo dell'estratto conto nell'aprile 2005 (dollaro NT)  
**X18** <- PAY\_AMT1: importo del pagamento precedente nel settembre 2005 (dollaro NT)  
**X19** <- PAY\_AMT2: importo del pagamento precedente nell'agosto 2005 (dollaro NT)  
**X20** <- PAY\_AMT3: importo del pagamento precedente nel luglio 2005 (dollaro NT)  
**X21** <- PAY\_AMT4: importo del pagamento precedente nel giugno 2005 (dollaro NT)  
**X22** <- PAY\_AMT5: importo del pagamento precedente a maggio 2005 (dollaro NT)  
**X23** <- PAY\_AMT6: importo del pagamento precedente nell'aprile 2005 (dollaro NT)  
**Y** <- default.payment.next.month: pagamento predefinito nel giugno 2005 (1 = sì, 0 = no)

## 5.4 Importazione dei dati

```
dati<-read.csv2("default_of_credit_card_clients.csv", h=T)
dati<- dati[-1,]
```

```
#importiamo il dataset di analisi rimuovendo la prima riga della
tabella in seguito alla modifica effettuata sui nomi delle variabili
```

## 5.5 Cleaning

```
str(dati)

#tutte le variabili vengono caricate come "character", dunque si rende
necessario convertire le stringhe in valori numeri

dati$X1<-as.numeric(as.character(dati$X1))
dati$X5<-as.numeric(as.character(dati$X5))
dati$X12<-as.numeric(as.character(dati$X12))
dati$X13<-as.numeric(as.character(dati$X13))
dati$X14<-as.numeric(as.character(dati$X14))
dati$X15<-as.numeric(as.character(dati$X15))
dati$X16<-as.numeric(as.character(dati$X16))
dati$X17<-as.numeric(as.character(dati$X17))
dati$X18<-as.numeric(as.character(dati$X18))
dati$X19<-as.numeric(as.character(dati$X19))
dati$X20<-as.numeric(as.character(dati$X20))
dati$X21<-as.numeric(as.character(dati$X21))
dati$X22<-as.numeric(as.character(dati$X22))
dati$X23<-as.numeric(as.character(dati$X23))
dati$Y<-as.numeric(as.character(dati$Y))
dati$Y<-as.factor(dati$Y)
dati$X2<-as.numeric(as.character(dati$X2))
dati$X2<-as.factor(dati$X2)

str(dati)
```

```
'data.frame': 30000 obs. of 25 variables:
 $ X : chr  "1" "2" "3" "4" ...
 $ X1 : num  20000 120000 90000 50000 50000 ...
 $ X2 : Factor w/ 2 levels "1","2": 2 2 2 2 1 1 1 1 2 2 1 ...
 $ X3 : chr  "2" "2" "2" "2" ...
 $ X4 : chr  "1" "2" "2" "1" ...
 $ X5 : num  24 26 34 37 57 37 29 23 28 35 ...
 $ X6 : chr  "2" "-1" "0" "0" ...
 $ X7 : chr  "2" "2" "0" "0" ...
 $ X8 : chr  "-1" "0" "0" "0" ...
 $ X9 : chr  "-1" "0" "0" "0" ...
 $ X10: chr  "-2" "0" "0" "0" ...
 $ X11: chr  "-2" "2" "0" "0" ...
 $ X12: num  3913 2682 29239 46990 8617 ...
 $ X13: num  3102 1725 14027 48233 5670 ...
 $ X14: num  689 2682 13559 49291 35835 ...
 $ X15: num  0 3272 14331 28314 20940 ...
 $ X16: num  0 3455 14948 28959 19146 ...
 $ X17: num  0 3261 15549 29547 19131 ...
 $ X18: num  0 0 1518 2000 2000 ...
 $ X19: num  689 1000 1500 2019 36681 ...
 $ X20: num  0 1000 1000 1200 10000 657 38000 0 432 0 ...
 $ X21: num  0 1000 1000 1100 9000 ...
 $ X22: num  0 0 1000 1069 689 ...
 $ X23: num  0 2000 5000 1000 679 ...
 $ Y : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 1 1 1 ...
```

Proseguiamo con il processo di pulizia dei dati. In particolare, osservando il dataset nel file .csv notiamo che ci sono alcuni valori delle variabili che sono distorti e presentano delle anomalie, che andremo ad aggiustare mediante una serie di cicli.

```
#Uniamo le classi della variabile EDUCATION (X3) che non rientrano nel range fissato dal testo

dati$X3<-as.numeric(as.character((dati$X3)))
for(i in 1:length(dati$X3)){
  if(dati$X3[i]==0) {dati$X3[i]<- "1"}
  else{if(dati$X3[i]>4){dati$X3[i]<-"4"}}
}
table(dati$X3)
dati$X3<-as.factor(dati$X3)

#Dobbiamo sistemare anche la variabile STATO CIVILE

dati$X4<-as.numeric(as.character(dati$X4))
for(i in 1:length(dati$X4)){
  if(dati$X4[i]==0) {dati$X4[i]<- "1"}
  else{if(dati$X4[i]>3){dati$X4[i]<-"3"}}
}
table(dati$X4)
dati$X4<-as.factor(dati$X4)

#Ora dobbiamo unire anche le modalità per le variabili X6, X7, X8, X9, X10, X11. Le osservazioni con modalità -2 verranno assegnate a -1

table(dati$X6)
dati$X6<-as.numeric(as.character(dati$X6))
for(i in 1:length(dati$X6)){
  if(dati$X6[i]<(1)) {dati$X6[i]<- "-1"}
}
table(dati$X6)
dati$X6<-as.factor(dati$X6)

dati$X7<-as.numeric(as.character(dati$X7))
for(i in 1:length(dati$X7)){
  if(dati$X7[i]<(1)) {dati$X7[i]<- "-1"}
}
table(dati$X7)
dati$X7<-as.factor(dati$X7)

dati$X8<-as.numeric(as.character(dati$X8))
for(i in 1:length(dati$X8)){
  if(dati$X8[i]<1) {dati$X8[i]<- "-1"}
}
table(dati$X8)
dati$X8<-as.factor(dati$X8)

dati$X9<-as.numeric(as.character(dati$X9))
for(i in 1:length(dati$X9)){
  if(dati$X9[i]<(1)) {dati$X9[i]<- "-1"}
}
table(dati$X9)
dati$X9<-as.factor(dati$X9)
```

```

dati$X10<-as.numeric(as.character(dati$X10))
for(i in 1:length(dati$X10)){
  if(dati$X10[i]<(1)) {dati$X10[i]<- "-1"}
}
table(dati$X10)
dati$X10<-as.factor(dati$X10)

dati$X11<-as.numeric(as.character(dati$X11))
for(i in 1:length(dati$X11)){
  if(dati$X11[i]<(1)) {dati$X11[i]<- "-1"}
}
table(dati$X11)

```

Questi aggiustamenti sono stati necessari in quanto, in alcune variabili, è stato opportuno unire delle categorie, in conseguenza al fatto che alcune modalità erano espresse con valori distorti rispetto al testo di riferimento. Per esempio la variabile Education, presentava delle modalità pari a 0 e pari a 4. Le modalità 0, sono state convertite in 1 e le modalità pari a 4, sono state convertite in 3. Discorso simile per la variabile Marriage in cui le osservazioni che presentavano modalità pari a 0, sono state convertite in 1 e quelle con modalità maggiore di 3, sono state accorpate con la modalità 3. Analogamente, per le variabili X6, X7, X8, X9, X10, X11, X12: le osservazioni che presentavano modalità uguale a 0, sono state convertite alla modalità -1.

## 5.6 Analisi delle variabili

Ora caricheremo una serie di librerie di R che ci saranno utili per il seguito dell'analisi esplorativa e la realizzazione di grafici dimostrativi.

```

library(MASS)
library(dplyr)
library(ggplot2)
library(gridExtra)

```

Iniziamo l'analisi esplorativa osservando la variabile default\_payment\_next\_month (Y) relativa ai clienti insolventi, e per prima cosa occorre vedere come è strutturata questa variabile.

```

default<-dati$Y
summary(default)

```

```
0   1  
23364 6636
```

```
#23364 osservazioni per default=0 (No) e 6636 osservazioni per  
default=1 (SI)
```

Su questa variabile bipartita, possiamo affermare che 23364 presentano modalità uguale a 0, cioè non vanno in default, mentre la restante parte (6636) ha valore uguale a 1, per cui va in default. In termini percentuali, il 78% della popolazione presa in esame non è insolvente.

Procediamo ora con una semplice valutazione della relazione tra questa variabile risposta e la variabile Sex, osservando la distribuzione di frequenza.

```
table(dati$X2,default)
```

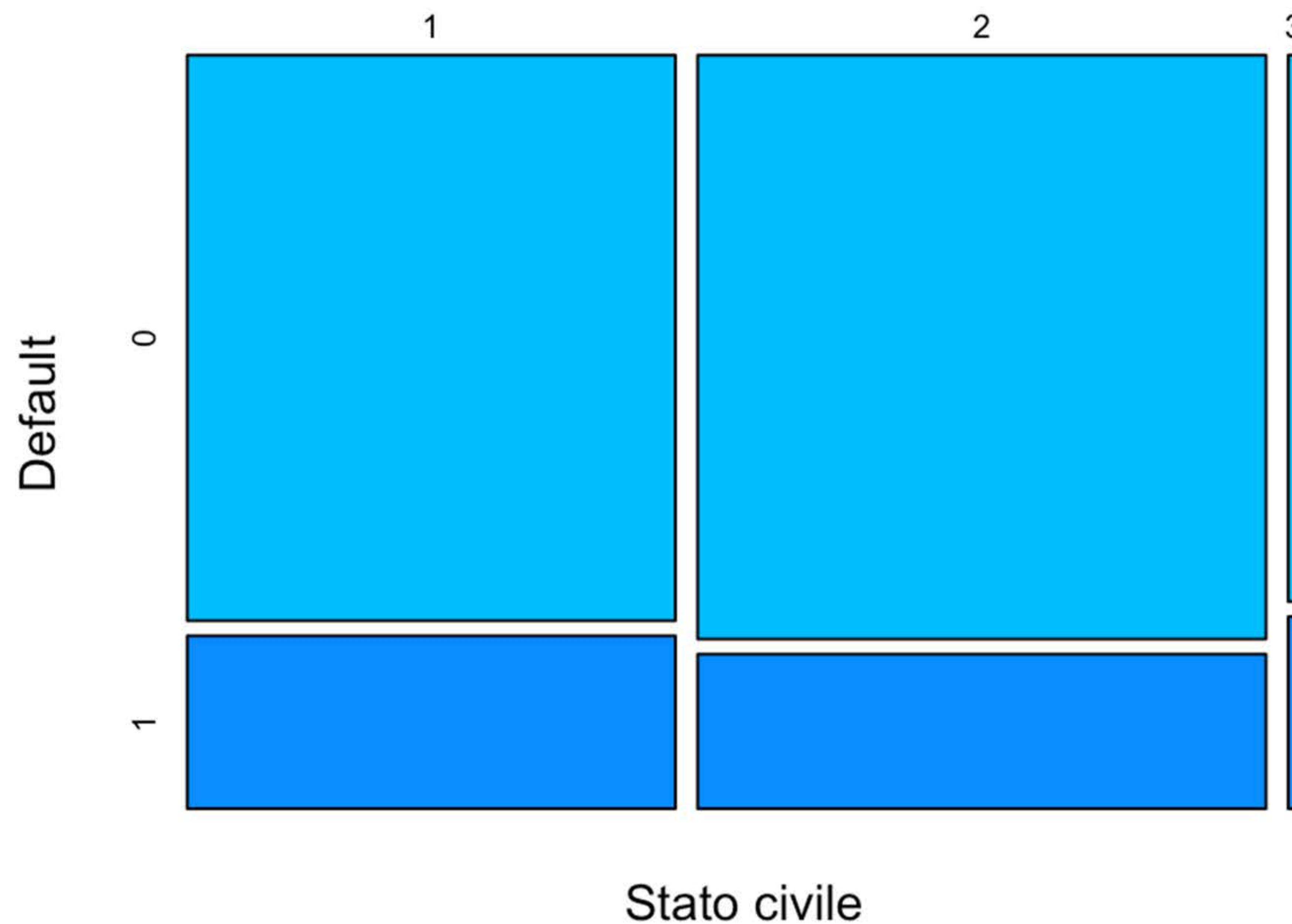
```
default  
0   1  
1  9015 2873  
2 14349 3763
```

Riportiamo questo risultato in una tabella per una migliore lettura dei dati.

Sex/Default	0	1	
Maschio	9015	2873	11888
Femmina	14349	3763	18112
Totale	23364	6636	30000

Dalla tabella è possibile valutare le percentuali delle due variabili. Si nota infatti che, tra i maschi, il 24% va in default, mentre le donne che vanno in default sono il 21%. Continuiamo l'analisi andando a valutare la variabile di default in relazione alla variabile Marriage, cioè lo stato civile dei clienti. Per questa valutazione useremo un grafico denominato *Mosaic plot*.

```
mosaicplot(structable(default~dati$X4, data=dati),  
           main =(""),  
           xlab=("Stato civile"),ylab=("Default"),  
           color=c("deepskyblue","dodgerblue1","dodgerblue2","dodgerblue3",  
           "dodgerblue4"))
```



Da quanto risulta dal mosaic plot, la variabile default non sembra essere particolarmente influenzata dalla variabile stato civile.

Procediamo confrontando la variabile default con le variabili che riguardano la storia del pagamento passato (X6, X7, X8, X9, X10, X11).

```

par(mfrow=c(3,2))

#Analisi tra default e X11(Aprile)
mosaicplot(structable(dati$Y~dati$X11, data=dati),
           main = "Aprile 2005",
           xlab=("Pagamento in ritardo"),ylab=("Default"),
           color=c("deepskyblue","dodgerblue1","dodgerblue2","dodgerblue3",
"dodgerblue4"))

#Analisi tra default e X10(Maggio)
mosaicplot(structable(dati$Y~dati$X10, data=dati),
           main = "Maggio 2005",
           xlab=("Pagamento in ritardo "),ylab=("Default"),
           color=c("deepskyblue","dodgerblue1","dodgerblue2","dodgerblue3",
"dodgerblue4"))

#Analisi tra default e X9(Giugno)
mosaicplot(structable(dati$Y~dati$X9, data=dati),
           main = "Giugno 2005",
           xlab=("Pagamento in ritardo"),ylab=("Default"),
           color=c("deepskyblue","dodgerblue1","dodgerblue2","dodgerblue3",
"dodgerblue4"))

```

```

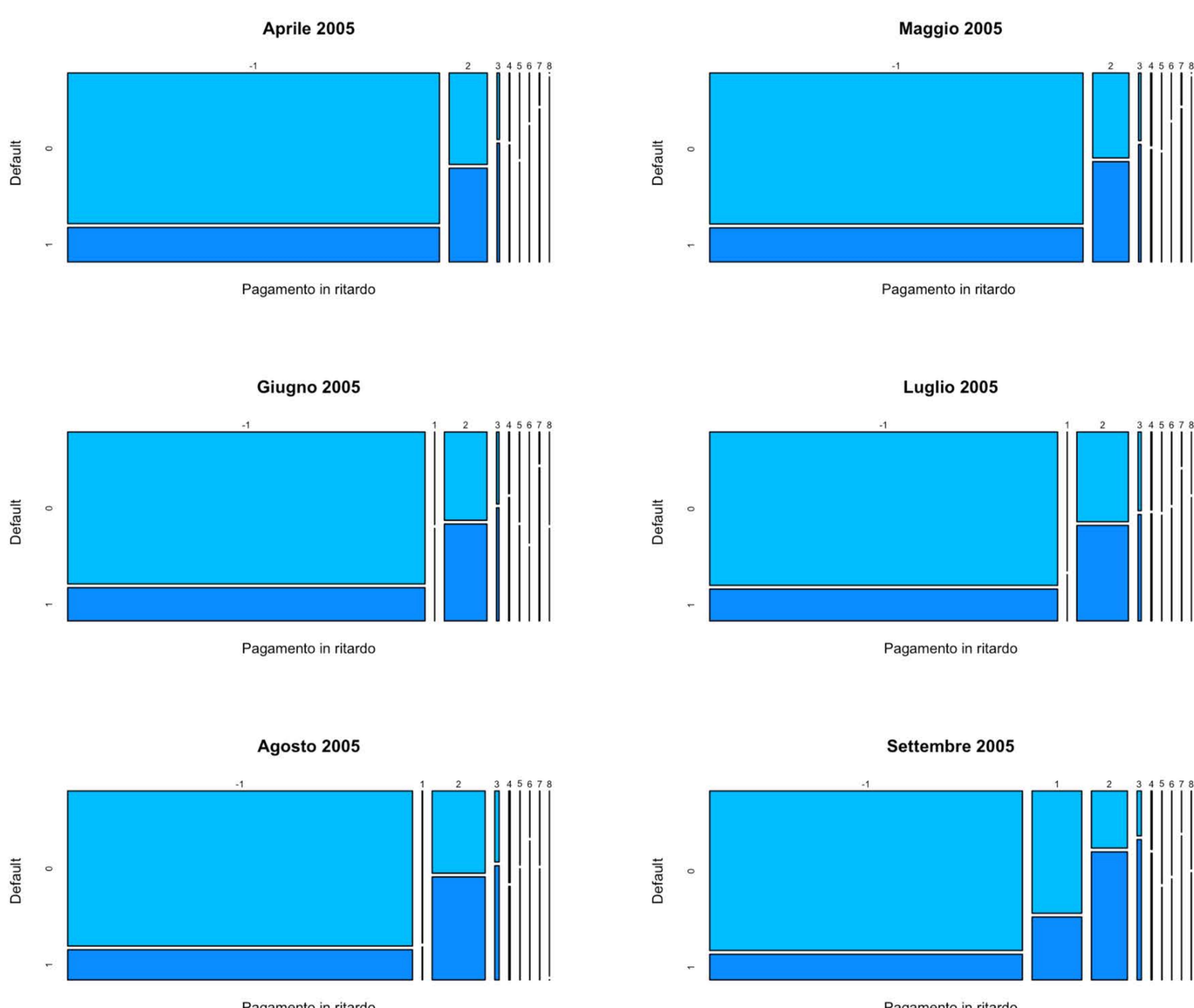
#Analisi tra default e X8(Luglio)
mosaicplot(structable(dati$Y~dati$X8, data=dati),
            main =("Luglio 2005"),
            xlab=("Pagamento in ritardo"),ylab=("Default"),
            color=c("deepskyblue","dodgerblue1","dodgerblue2","dodgerblue3",
            "dodgerblue4"))

#Analisi tra default e X7(Agosto)
mosaicplot(structable(dati$Y~dati$X7, data=dati),
            main = "Agosto 2005",
            xlab=("Pagamento in ritardo"),ylab=("Default"),
            color=c("deepskyblue","dodgerblue1","dodgerblue2","dodgerblue3",
            "dodgerblue4"))
#Quasi nessuno paga con un mese di ritardo.

#Analisi tra default e X6(Settembre)
mosaicplot(structable(dati$Y~dati$X6, data=dati),
            main = "Settembre 2005",
            xlab=("Pagamento in ritardo"),ylab=("Default"),
            color=c("deepskyblue","dodgerblue1","dodgerblue2","dodgerblue3",
            "dodgerblue4"))

#si nota che l'andamento di andare in default, aumenta man mano
che si paga in ritardo, fino a 3 mesi

```



Dal grafico si evince che chi paga puntuale nei mesi presi in esame, ha più o meno la stessa probabilità di andare in default. Si nota inoltre che oltre al mese di Settembre, vi sono poche osservazioni in cui la popolazione presa in esame paga con un mese di ritardo. Una buona parte delle osservazioni paga con due mesi di ritardo nei sei mesi presi in esame ed è possibile valutare che la probabilità di andare in default per chi paga con due mesi di ritardo, è pressoché simile nei mesi, tranne per il mese di Settembre. Si evince infatti che chi paga con due mesi di ritardo nel mese di Settembre, andrà in default in proporzioni maggiori. Si vede anche che le osservazioni che pagano con più di 3 mesi di ritardo, sono poche, ma si nota che esse andranno in default più facilmente rispetto a chi paga puntuale.

## 5.7 Boxplot condizionati

In statistica il *boxplot* o *diagramma a scatola e baffi*, è una rappresentazione grafica utilizzata per descrivere la distribuzione di un campione tramite semplici indici di dispersione e di posizione<sup>11</sup>. Presenta fondamentalmente 5 valutazioni di sintesi:

- > il valore minimo (min)
- > il primo quartile (Q1)
- > la mediana (o secondo quartile, Q2)
- > il terzo quartile (Q3)
- > il valore massimo (max)

Concludiamo la trattazione andando ad analizzare la variabile default con le variabili che riguardano l'ammontare dell'estratto conto nei mesi di riferimento (X12, X13, X14, X15, X16, X17), tenendo sempre in considerazione la relazione con lo stato di pagamento in ritardo, visto in precedenza.

Per quest'ultima analisi useremo una serie di boxplot condizionati che ci daranno una rappresentazione di questa relazione, nei mesi da Aprile a Settembre 2005.

```
par(mfrow=c(3,2))

(a<-ggplot(data=dati) +
 geom_boxplot(mapping=aes(x=dati$X11,y=dati$X17,col=dati$Y)) +
 labs(
   title = "Aprile 2005",
   x = "Pagamento in ritardo",
   y = "Ammontare dell'estratto conto")+
 scale_colour_brewer(palette = "Set2")+
 theme(legend.position = "bottom")
```

<sup>11</sup> [https://it.wikipedia.org/wiki/Diagramma\\_a\\_scatola\\_e\\_baffi](https://it.wikipedia.org/wiki/Diagramma_a_scatola_e_baffi)

```

(b<-ggplot(data=dati) +
  geom_boxplot(mapping=aes(x=dati$X10,y=dati$X16,col=dati$Y)) +
  labs(
    title = "Maggio 2005",
    x = "Pagamento in ritardo",
    y = "Ammontare dell'estratto conto")+
  scale_colour_brewer(palette = "Set2")+
  theme(legend.position = "bottom"))


(c<-ggplot(data=dati) +
  geom_boxplot(mapping=aes(x=dati$X9,y=dati$X15,col=dati$Y)) +
  labs(
    title = "Giugno 2005",
    x = "Pagamento in ritardo",
    y = "Ammontare dell'estratto conto")+
  scale_colour_brewer(palette = "Set2")+
  theme(legend.position = "bottom"))

(d<-ggplot(data=dati) +
  geom_boxplot(mapping=aes(x=dati$X8,y=dati$X14,col=dati$Y)) +
  labs(
    title = "Luglio 2005",
    x = "Pagamento in ritardo",
    y = "Ammontare dell'estratto conto")+
  scale_colour_brewer(palette = "Set2")+
  theme(legend.position = "bottom"))

(e<-ggplot(data=dati) +
  geom_boxplot(mapping=aes(x=dati$X7,y=dati$X13,col=dati$Y)) +
  labs(
    title = "Agosto 2005",
    x = "Pagamento in ritardo",
    y = "Ammontare dell'estratto conto")+
  scale_colour_brewer(palette = "Set2")+
  theme(legend.position = "bottom"))

(f<-ggplot(data=dati) +
  geom_boxplot(mapping=aes(x=dati$X6,y=dati$X12,col=dati$Y)) +
  labs(
    title = "Settembre 2005",
    x = "Pagamento in ritardo",
    y = "Ammontare dell'estratto conto")+
  scale_colour_brewer(palette = "Set2")+
  theme(legend.position = "bottom"))

```



Da tutti i grafici si nota una situazione comune, cioè, per chi paga puntuale, l'ammontare dell'estratto conto è mediamente più basso, rispetto a quello di chi paga con almeno 4 mesi di ritardo e ovviamente anche in questo caso si evidenzia che chi paga in ritardo, ha una tendenza maggiore di andare in default, rispetto a chi paga puntuale.

Con questo ultimo punto concludiamo la nostra EDA sul dataset presentato.

A titoli di commento, osserviamo infine che una possibile estensione di questo modello, potrebbe consistere nell'applicazione di un algoritmo di *Random Forest*. Si tratta di una delle metodologie di *machine learning* più utilizzate, e consiste in un algoritmo di apprendimento, ottenuto dall'aggregazione tramite *bagging* di alberi di decisione. La tecnica di Random Forest si pone come soluzione che minimizza l'*overfitting* del *training set* rispetto agli alberi di decisione.

## 6. Bibliografia e referenze

<https://cran.r-project.org/> - "Una guida all'utilizzo dell'ambiente statistico R" by Angelo M. Mineo, an introductory guide, based mainly on "An introduction to R"

<https://cran.r-project.org/> - "Il linguaggio R: concetti introduttivi ed esempi" (II edizione) by Vito M. R. Muggeo and Giancarlo Ferrara

<https://www.html.it/guide/guida-r/>

<https://www.html.it/pag/65133/data-processing-con-r/>

<https://www.r-exercises.com/start-here-to-learn-r>

<https://github.com/Avani10/Taiwan-Credit-Default-Analysis-in-R>

<https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>

[https://github.com/Avani10/Taiwan-Credit-Default-Analysis-in-R/blob/master/Data/Taiwan\\_credit\\_default\\_data\\_final\\_v1.csv](https://github.com/Avani10/Taiwan-Credit-Default-Analysis-in-R/blob/master/Data/Taiwan_credit_default_data_final_v1.csv)

<https://it.wikipedia.org/>