Progetto Biostatistica

Mirco Durante 829408, Giovanni Cornacchia 830631

2022-09-13

Contents

Introduzione

Per il progetto di Laboratorio per la Biostatistica il nostro obiettivo è stato quello di partire da un dataset contenente errori e ripurirlo da essi attraverso la procedura di cleaning. Una volta effettuato questo passaggio abbiamo cercato di dare qualche informazione sulle variabili e provato a svolgere un'analisi preliminare di sopravvivenza.

Dataset

Il dataset è denominato SOFA. Il punteggio SOFA valuta la disfunzione d'organo, in letteratura è noto, come un incremento del punteggio, aumenti il rischio di mortalità.

Le variabili di cui si compone il set di dati in questione sono le seguenti:

1. **PAZIENTE**: *numeric*,numero identificativo del paziente
2. **NASCITA**: *character*,data di nascita del paziente
3. **SEX**: *numeric*,sesso del paziente
4. **STATCIV**: *numeric*,stato civile del paziente
5. **PESO**: *numeric*,peso del paziente arrotondato in kg
6. **ALTEZ**: *numeric*,altezza del paziente in cm
7. **CADUTE**: *numeric*,numero di cadute del paziente
8. **CCSCORE**: *numeric*,Charlson comorbidity index
9. **SOFAING**: *numeric*, punteggio Sofa
10. **MMSE**: *character*, Mini-Mental State Examination
11. **ALB**: *character*,valore di albumina
12. **CALC**: *character*,valore di calcio
13. **VITD**: *character*,valore di vitamina D
14. **HBING**: *character*,valore dell'emoglobina
15. **TEMPRIC**: *numeric*,tempo di ricovero in minuti
16. **DATDIM**: *POSIXct*,data della dimissione del paziente dall'ospedale
17. **DATINT**: *POSIXct*,data dell'intervento
18. **INTDURAT**: *numeric*,durata dell'intervento in minuti
19. **ANEST**: *numeric*,tipo di anestesia utilizzata
20. **DATA DECESSO**: *character*,data decesso del paziente

Librerie e Funzioni

```
library(readxl)
#install.packages("stringr")
library("stringr")
require(lubridate)
#install.packages("visdat")
library(MASS)
library(visdat)
library(lattice)
library(ggplot2)
library(ggrepel)
library(survival)
library(ggsurvey)
```

```
library(DataExplorer)
library(RColorBrewer)
library(ggplot2)
library(ggfortify)
library(plotly)
library(survminer)
#install.packages("epitools")
library(epitools)
library(ggfortify)
library(MatchIt)
library(table1)
library('psych')
library(ggcorrplot)
library(MatchIt)
library(table1)
#install.packages("DataExplorer")
library(DataExplorer)
library(RColorBrewer)
library(ggplot2)
library(ggfortify)
library(plotly)
library(survminer)
library(ggfortify)
library(rms)
#install.packages("vioplot")
library("vioplot")

## funzione per accentrare il titolo
custom_theme <- function() {
  theme_survminer() %+replace%
    theme(
      plot.title=element_text(hjust=0.5)
    )
}
```

Data Acquisition

```
setwd("C:/Users/Mirco/Desktop/Uni/Biostatistica/Laboratorio R per la Biostatistica")
SOFA <- read_excel("SOFA.xlsx")
#View(SOFA)
```

Data Esploration

Andiamo ad osservare tutte le variabili presenti nel dataset e le relative classi.

```
str(SOFA)
```

```
## tibble [64 x 20] (S3: tbl_df/tbl/data.frame)
##  $ PAZIENTE    : num [1:64] 1 2 5 11 14 16 16 20 22 23 ...
```

```
##  $ NASCITA     : chr [1:64] "14816" "10527" "10264" "8085" ...
##  $ SEX         : num [1:64] 1 2 1 2 2 2 2 1 2 1 ...
##  $ STATCIV     : num [1:64] 2 2 2 5 5 5 2 2 2 2 ...
##  $ PESO        : num [1:64] 75 40 70 45 67 50 57 85 60 62 ...
##  $ ALTEZ       : num [1:64] 170 135 170 158 170 150 165 180 165 179 ...
##  $ CADUTE      : num [1:64] 1 2 1 1 2 2 1 1 2 2 ...
##  $ CCSCORE     : num [1:64] 5 2 3 3 0 3 5 3 4 3 ...
##  $ SOFAING     : num [1:64] 0 2 2 0 0 0 0 1 0 3 ...
##  $ MMSE        : chr [1:64] "22,3" "27.7" "0" "17.399999999999999" ...
##  $ ALB         : chr [1:64] "3,6" "3.37" "3" "2.6" ...
##  $ CALC        : chr [1:64] "9,9" "9.6" "9" "7.3" ...
##  $ VITD        : chr [1:64] "-1" "3.1" "9.1" "3" ...
##  $ HBING       : chr [1:64] "13,2" "14.1" "13" "13.3" ...
##  $ TEMPRIC     : num [1:64] 55 60 120 60 144 120 220 450 200 270 ...
##  $ DATDIM      : POSIXct[1:64], format: "2011-11-02" NA ...
##  $ DATINT      : POSIXct[1:64], format: "2011-10-19" "2012-01-13" ...
##  $ INTDURAT    : num [1:64] 40 95 35 100 80 55 140 100 50 75 ...
##  $ ANEST       : num [1:64] 2 2 4 5 1 1 1 4 4 1 ...
##  $ DATA DECESSO: chr [1:64] NA "40927" "41010" "40947" ...
```

Data Cleaning

Nella fase di data cleaning andremo ad osservare per ogni variabile se vi siano delle incongruenze nei valori e cercheremo di risolvere questi problemi rendendo così il dataset più pulito per le fasi di analisi successive.

PAZIENTE

Controlliamo che non vi siano dublicati nei pazienti.

```
SOFA$PAZIENTE[duplicated(SOFA$PAZIENTE)]
```

```
## [1] 16
```

```
SOFA[SOFA$PAZIENTE== 16,]
```

```
## # A tibble: 2 x 20
##   PAZIENTE NASCITA   SEX STATCIV  PESO ALTEZ CADUTE CCSCORE SOFAING MMSE  ALB
##      <dbl> <chr>   <dbl>   <dbl> <dbl> <dbl>  <dbl>   <dbl>   <dbl> <chr> <chr>
## 1       16 6967        2       5    50   150      2       3       0 0     3,85
## 2       16 6967        2       2    57   165      1       5       0 25,1  4,1
## # ... with 9 more variables: CALC <chr>, VITD <chr>, HBING <chr>,
## #   TEMPRIC <dbl>, DATDIM <dttm>, DATINT <dttm>, INTDURAT <dbl>, ANEST <dbl>,
## #   `DATA DECESSO` <chr>
```

Il valore id 16 si ripete 2 volte. Osservando che le variabili riferite ai due pazienti sono diverse tra loro ad eccezione della data di nascita, possiamo affermare che i due siano due persone diverse. Trattiamo questo errore cambiando il codice identificativo di uno dei due pazienti con un numero che non è presente nella lista, in modo tale da non perdere l'informazione.

Assegnamo al secondo paziente con id 16 posto in riga 7, il valore 17.

```
SOFA[7,1] = 17
SOFA[7,1]
```

```
## # A tibble: 1 x 1
##   PAZIENTE
##      <dbl>
## 1       17
```

```
SOFA[SOFA$PAZIENTE==16 | SOFA$PAZIENTE==17,]
```

```
## # A tibble: 2 x 20
##   PAZIENTE NASCITA   SEX STATCIV  PESO ALTEZ CADUTE CCSCORE SOFAING MMSE  ALB
##      <dbl> <chr>   <dbl>   <dbl> <dbl> <dbl>  <dbl>   <dbl>   <dbl> <chr> <chr>
## 1       16 6967        2       5    50   150      2       3       0 0     3,85
## 2       17 6967        2       2    57   165      1       5       0 25,1  4,1
## # ... with 9 more variables: CALC <chr>, VITD <chr>, HBING <chr>,
## #   TEMPRIC <dbl>, DATDIM <dttm>, DATINT <dttm>, INTDURAT <dbl>, ANEST <dbl>,
## #   'DATA DECESSO' <chr>
```

NASCITE

Osservando le date di nascita dei vari pazienti troviamo molti problemi. Dapprima si nota come le date non siano in formato classico dd/mm/yyyy ma sono numeri interi classificati come carattere. Notiamo inoltre 3 dati diversi rispetto agli altri. 2 sono in formato dd/mm/yyyy ma non sono stati convertiti in numero e ad 1 manca uno '/' che separa i mesi dagli anni. Andando con ordine, modifichiamo il dato relativo al paziente 111 aggiungendo uno '/' che separa mesi da anni.

```
(i1 <- SOFA[SOFA$PAZIENTE==111,])
SOFA$NASCITA[SOFA$PAZIENTE==111]=sapply(Map(append, strsplit(i1$NASCITA,""), after = nchar(i1$NASCITA) -
SOFA[SOFA$PAZIENTE==111,]
```

Andiamo poi ad evidenziare le date che hanno un formato differente rispetto alle altre:

```
SOFA$NASCITA[c(43, 61, 34)]
```

```
## [1] "16/03/1928" "04/07/1822" "13/07/1829"
```

transformiamo i numeri in date consideranto come data di origine il 1899-12-30

```
SOFA_1 = SOFA
SOFA_1$NASCITA = as.numeric(SOFA$NASCITA)
(SOFA_1$NASCITA = as.Date(x = SOFA_1$NASCITA,origin = "1899-12-30", ))
```

```
##  [1] "1940-07-24" "1928-10-26" "1928-02-06" "1922-02-18" "1924-04-12"
##  [6] "1919-01-27" "1919-01-27" "1916-10-07" "1935-01-28" "1924-05-20"
## [11] "1937-09-12" "1909-01-07" "1922-01-13" "1928-04-02" "1924-02-12"
## [16] "1931-10-15" "1932-09-25" "1933-11-13" "1928-10-10" "1930-12-07"
## [21] "1925-01-06" "1927-07-01" "1922-08-31" "1914-03-24" "1919-12-12"
## [26] "1921-11-07" "1927-11-13" "1929-12-18" "1922-03-26" "1930-02-28"
## [31] "1931-05-17" "1916-08-25" "1914-07-12" NA           "1937-08-01"
```

```
## [36] "1931-02-28" "1920-07-26" "1932-09-13" "1928-06-30" "1920-01-14"
## [41] "1928-06-25" "1924-09-16" NA           "1929-09-15" "1911-10-15"
## [46] "1926-11-06" "1927-03-10" "1936-01-28" "1924-03-01" "1928-04-27"
## [51] "1932-12-19" "1926-07-17" "1925-08-17" "1925-01-30" "1922-06-15"
## [56] "1923-07-05" "1926-11-28" "1941-05-23" "1932-08-16" "1921-02-05"
## [61] NA           "1922-03-10" "1926-08-02" "1922-01-13"
```

Nelle celle relative alle date diverse evidenziate prima, vengono posti di default degli NA. Possiamo dunque sostituire i valori nulli con le 3 date modificate.

Ponendo l'attenzione sulle date con anni 1822 e 1829 e osservando le date di intervento, ci sembra improbabile che esse siano corrette. Pensiamo dunque che ci sia stato un errore di battitura nella compilazione del dataset. A questo proposito ci sembra più ragionevole modificare gli anni ponendo 1922 e 1929.

```
SOFA_1$NASCITA[c(43, 61, 34)] = as.Date(c("1928/03/16","1922/07/04","1929/07/13"))
```

SEX

Per la variabile SEX relativa alla differenza di genere non troviamo problemi ne valori missing. Creiamo una variabile denominata sex_lev rinominando i fattori in 'uomo' quando sex=1 e 'donna' quando sex=2.

```
#sex

SOFA_1$SEX = as.factor(SOFA_1$SEX)
SOFA_1$sex_lev[SOFA_1$SEX== 1]='uomo'
SOFA_1$sex_lev[SOFA_1$SEX== 2]='donna'
#View(SOFA_1)
```

STATCIV

Lo Stato civile del paziente non trova errori ne missing. Creiamo una nuova variabile "STATCIV_LEV" con le varie classi.

```
#STATCIV
SOFA_1$STATCIV = as.factor(SOFA_1$STATCIV)

SOFA_1$STATCIV_lev[SOFA_1$STATCIV== 1]='non sposato'
SOFA_1$STATCIV_lev[SOFA_1$STATCIV== 2]='congiunto/a'
SOFA_1$STATCIV_lev[SOFA_1$STATCIV== 3]='convivente'
SOFA_1$STATCIV_lev[SOFA_1$STATCIV== 4]='separato/a'
SOFA_1$STATCIV_lev[SOFA_1$STATCIV== 5]='vedovo/a'
```

PESO

```
SOFA_1$PESO
```

```
##  [1] 75 40 70 45 67 50 57 85 60 62 68 60 65 63 70 65 64 53 80 64 54 80 NA 45 70
## [26] 55 65 70 95 55 65 40 50 50 80 90 86 77 70 40 70 -1 68 50 70 59 94 57 64 65
## [51] -1 50 33 45 50 39 50 40 50 50 48 60 57 50
```

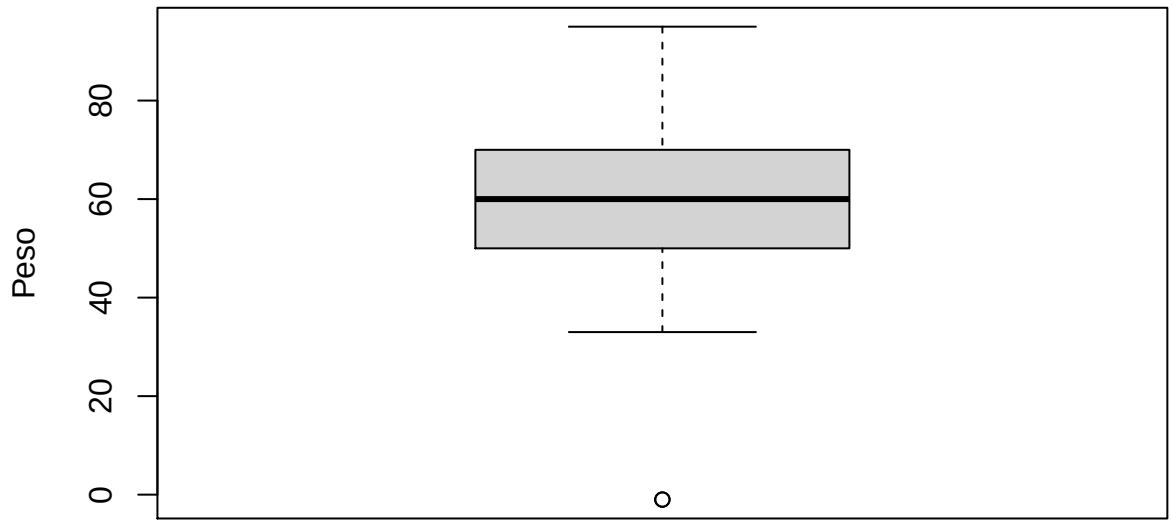
Il peso presenta dei valori errati pari a -1 che in fase di analisi potrebbero creare problemi. Inoltre vi è un paziente con un peso di 33 kg. Per questo dato possiamo fare una piccola analisi individuando se sia un potenziale outlier attraverso un boxplot.

```
SOFA_1[SOFA_1$PESO==33,]
```

```
## # A tibble: 2 x 22
##   PAZIENTE NASCITA    SEX   STATCIV  PESO ALTEZ CADUTE CCSCORE SOFAING MMSE
##      <dbl> <date>     <fct> <fct>   <dbl> <dbl>  <dbl>   <dbl>   <dbl> <chr>
## 1       NA NA         <NA>  <NA>       NA    NA     NA      NA      NA <NA>
## 2      141 1925-08-17 2     5          33   160      2       3       0 18.39999~
## # ... with 12 more variables: ALB <chr>, CALC <chr>, VITD <chr>, HBING <chr>,
## #   TEMPRIC <dbl>, DATDIM <dttm>, DATINT <dttm>, INTDURAT <dbl>, ANEST <dbl>,
## #   'DATA DECESSO' <chr>, sex_lev <chr>, STATCIV_lev <chr>
```

boxplot peso

```
p = boxplot(SOFA_1$PESO, ylab = 'Peso')
```




```
boxplot.stats(SOFA_1$PESO)$out
```

```
## [1] -1 -1
```

Dal grafico si evidenzano 2 outlier (nello stesso punto -1), siamo quindi propensi a mantenere il dato di peso 33.

Sostituiamo quindi i valori negativi con NA.

```
SOFA_1$PESO =ifelse(SOFA_1$PESO==-1,NA,SOFA_1$PESO)
SOFA_1$PESO
```

```
##  [1] 75 40 70 45 67 50 57 85 60 62 68 60 65 63 70 65 64 53 80 64 54 80 NA 45 70
## [26] 55 65 70 95 55 65 40 50 50 80 90 86 77 70 40 70 NA 68 50 70 59 94 57 64 65
## [51] NA 50 33 45 50 39 50 40 50 50 48 60 57 50
```

```
min(na.omit(SOFA_1$PESO))
```

```
## [1] 33
```

```
max(na.omit(SOFA_1$PESO))
```

```
## [1] 95
```

Facendo un rapido controllo possiamo notare come il valore minimo sia 33 e il massimo 95 e non ci siano altri problemi.

ALTEZZA

Nelle altezze troviamo anche qui qualche incongruenza.

```
SOFA_1$ALTEZ
```

```
##  [1] 170 135 170 158 170 150 165 180 165 179 158 160 150 165 168  NA 160 169 170
## [20] 161 160 150 164 150 154 155 155 156 173 154 165 150 165 155 165 165 160 159
## [39] 160 150 170  -1 160 150 167 155 165 160 165 168   9 160 160 155 160 150 160
## [58] 165 150 170 150 160 150 150
```

```
min(na.omit(SOFA_1$ALTEZ))
```

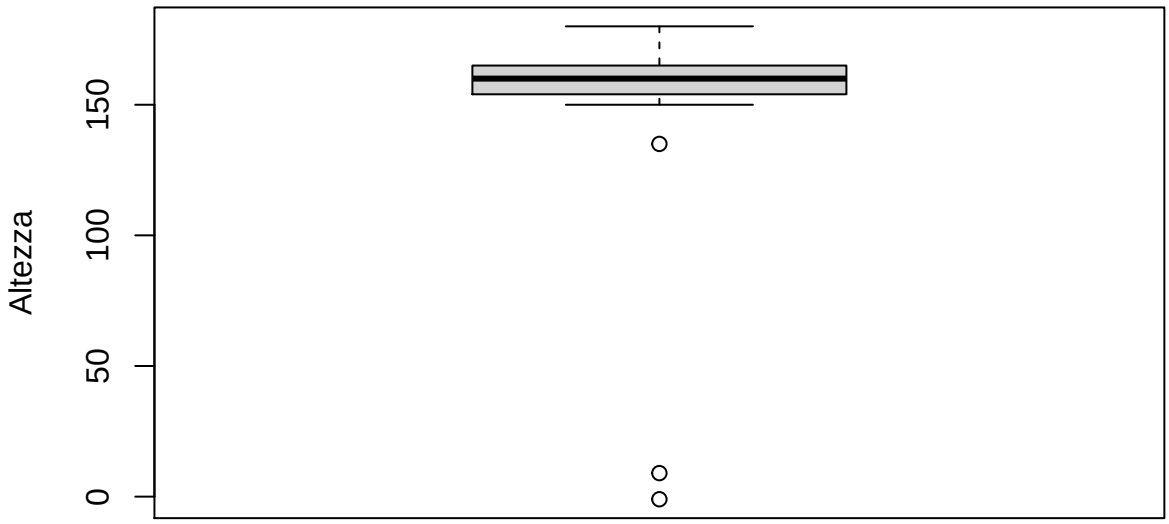
```
## [1] -1
```

```
max(na.omit(SOFA_1$ALTEZ))
```

```
## [1] 180
```

Osservando valori minimi e massimo, escludendo gli NA troviamo valori minimi anomali. Cerchiamo inoltri altri potenziali outlier.

```
p = boxplot(SOFA_1$ALTEZ, ylab = 'Altezza')
```




```
boxplot.stats(SOFA_1$ALTEZ)$out
```

```
## [1] 135  -1   9
```

Ci vengono identificati come possibili outlier i 2 valori negativi che andremo a sostituire con valori missing.

```
SOFA_1$ALTEZ = ifelse(SOFA_1$ALTEZ< 100,NA,SOFA_1$ALTEZ)
SOFA_1$ALTEZ
```

```
##  [1] 170 135 170 158 170 150 165 180 165 179 158 160 150 165 168  NA 160 169 170
## [20] 161 160 150 164 150 154 155 155 156 173 154 165 150 165 155 165 165 160 159
## [39] 160 150 170  NA 160 150 167 155 165 160 165 168  NA 160 160 155 160 150 160
## [58] 165 150 170 150 160 150 150
```

CADUTE

La variabile CADUTE indica il **numero di cadute** registrate per ogni paziente.

```
max(SOFA_1$CADUTE)
```

```
## [1] 2
```

```
min(SOFA_1$CADUTE)
```

```
## [1] -1
```

```
SOFA_1$CADUTE = ifelse(SOFA_1$CADUTE<0,NA,SOFA_1$CADUTE)
min(SOFA_1$CADUTE)
```

```
## [1] NA
```

Dato il valore minimo della variabile -1,non realizzabile per una varibile che spiega il numero di cadute registrate, poniamo il dato come valore missing.

CCSCORE

Charlson comorbidity index è un punteggio che viente attribuito ad ogni paziente in base alle comorbidità in esso presenti. Si determinano 7 livelli, dal punteggio più basso 0 a quello più alto 6.

```
SOFA_1$CCSCORE = as.factor(SOFA_1$CCSCORE)
table(SOFA_1$CCSCORE)
```

```
##
##  0  1  2  3  4  5  6
## 16  4  9 14  9  8  4
```

Quello che possiamo fare per rendere più leggibile il nostro dataset è quello di creare una nuova variabile indicando la classe del punteggio ccscore. Abbiamo optato per dividere in 3 livelli il punteggio.

- ccscore = 0,1,2 –> ccscore basso
- ccscore = 3,4 –> ccscore medio
- ccscore = 5,6 –> ccscore alto

```
SOFA_1$CCSCORE<-as.numeric(SOFA_1$CCSCORE)
SOFA_1$CCSCORE_cla = ifelse((SOFA_1$CCSCORE==0 |SOFA_1$CCSCORE==1 |SOFA_1$CCSCORE==2),'ccscore basso',
                        ifelse((SOFA_1$CCSCORE==3|SOFA_1$CCSCORE==4),'ccscore medio','ccscore alto'))
(table(SOFA_1$CCSCORE_cla))
```

```
##
##  ccscore alto ccscore basso ccscore medio
##            21            20            23
```

SOFAING

La variabili SOFAING indica il punteggio SOFA, che come abbiamo anticipato, valuta la **disfunzione d'organo**. In questo caso viene classificato su 5 livelli, dal punteggio più basso 0 a quello più alto 4.

```
SOFA_1$SOFAING = as.factor(SOFA_1$SOFAING)
table(SOFA_1$SOFAING)
```

```
##
##  0  1  2  3  4
## 44 11  6  2  1
```

Anche per questa variabile abbiamo creato delle classi dividendo in:

- sofaing = 0 –> sofa basso
- sofaing = 1,2,3,4 —> sofa alto

```
SOFA_1$sofa_class = ifelse(SOFA_1$SOFAING== '0','Sofa Basso', 'Sofa Alto')
table(SOFA_1$sofa_class)
```

```
##
##  Sofa Alto Sofa Basso
##         20         44
```

Mini-Mental State Examination (MMSE)

Il MMSE rappresenta un **rapido e sensibile strumento per l'esplorazione della funzione cognitiva e delle sue modificazioni nel tempo**. Viene spesso utilizzato come strumento di screening nell'indagine di soggetti con demenza e con sindromi neuropsicologiche di natura differente.

Osserviamo che per alcuni dati viene segnato il valore -1 indicando il dato mancante. Al fine di avere una più corretta interpretazione successiva poniamo questi valori come missing.

```
SOFA_1$MMSE
```

```
##  [1] "22,3"               "27.7"               "0"
##  [4] "17.399999999999999" "29,8"               "0"
##  [7] "25,1"               "5,4"                "4.7"
## [10] "27.3"               "14,3"               "0"
## [13] "23.8"               "24,1"               "28.8"
## [16] "12.7"               "27.7"               "27"
## [19] "21,1"               "0"                  "16,4"
## [22] "23.5"               "19.399999999999999" "0"
## [25] "13.4"               "26.4"               "17,4"
## [28] "21,4"               "11.4"               "7,1"
## [31] "23,1"               "13.4"               "-1"
## [34] "20,4"               "16,4/25"            "30"
## [37] "26.4"               "27.7"               "26.4"
## [40] "12.4"               "18,4"               "9.4"
## [43] "4,4"                "4,4"                "15,4"
## [46] "24.4"               "25,7"               "25.7"
## [49] "30,4"               "24.1"               "14.7"
## [52] "6"                  "18.399999999999999" "0"
## [55] "21.2"               "-1"                 "27.8"
## [58] "30"                 "14"                 "21"
## [61] "18.399999999999999" "25.2"               "0"
## [64] "0"
```

```
min(na.omit(SOFA_1$MMSE))
```

```
## [1] "-1"
```

```
SOFA_1$MMSE = ifelse(SOFA_1$MMSE<0,NA,SOFA_1$MMSE)
SOFA_1$MMSE
```

```
##  [1] "22,3"               "27.7"               "0"
##  [4] "17.399999999999999" "29,8"               "0"
##  [7] "25,1"               "5,4"                "4.7"
## [10] "27.3"               "14,3"               "0"
## [13] "23.8"               "24,1"               "28.8"
## [16] "12.7"               "27.7"               "27"
## [19] "21,1"               "0"                  "16,4"
## [22] "23.5"               "19.399999999999999" "0"
## [25] "13.4"               "26.4"               "17,4"
## [28] "21,4"               "11.4"               "7,1"
## [31] "23,1"               "13.4"               NA
## [34] "20,4"               "16,4/25"            "30"
## [37] "26.4"               "27.7"               "26.4"
## [40] "12.4"               "18,4"               "9.4"
## [43] "4,4"                "4,4"                "15,4"
## [46] "24.4"               "25,7"               "25.7"
## [49] "30,4"               "24.1"               "14.7"
## [52] "6"                  "18.399999999999999" "0"
## [55] "21.2"               NA                   "27.8"
## [58] "30"                 "14"                 "21"
## [61] "18.399999999999999" "25.2"               "0"
## [64] "0"
```

Notamo anche che alcuni valori presentino virgole "," al posto di punti "." quindi R non li riconosce come valori decimali. Sostituiamo dunque con il punto la virgola.

```
SOFA_1$MMSE = str_replace(SOFA_1$MMSE, ",", ".")
SOFA_1$MMSE = as.numeric(SOFA_1$MMSE)
```

ALB

L'albumina è considerata una delle proteine più importanti dell'organismo ed è contenuta soprattutto nel plasma. Nel nostro dataset il valore -1 indica il valore mancante perciò lo sostituiamo come NA. Sostituiamo inoltre le virgole con i punti.

```
SOFA_1$ALB
```

```
##  [1] "3,6"  "3.37" "3"    "2.6"  "3,1"  "3,85" "4,1"  "3,0"  "2.8"  "3.1"
## [11] "3,39" "3.4"  "3.55" "3,58" "3.86" "3.6"  "4.5"  "3,98" "3,49" "3.26"
## [21] "3,89" "3.36" "3.76" "-1"   "3.09" "2.88" "4,2"  "3,74" "3.17" "4"
## [31] "3,78" "3.4"  "2,78" "2,81" "4,3"  "3.16" "3"    "3.44" "2.9"  "2.84"
## [41] "3,7"  "3.1"  "3,05" "3,3"  "3,32" "3.06" "3,1"  "3.65" "3,6"  "3.5"
## [51] "2.95" "3.6"  "3.67" "3.23" "3.43" "3,2"  "3.54" "2.68" "3,87" "3,25"
## [61] "3.5"  "3.35" "3.6"  "2.56"
```

```
SOFA_1$ALB = ifelse(SOFA_1$ALB==-1,NA,SOFA_1$ALB)
SOFA_1$ALB = str_replace(SOFA_1$ALB, ",", ".")
SOFA_1$ALB = as.numeric(SOFA_1$ALB)
SOFA_1$ALB
```

```
##  [1] 3.60 3.37 3.00 2.60 3.10 3.85 4.10 3.00 2.80 3.10 3.39 3.40 3.55 3.58 3.86
## [16] 3.60 4.50 3.98 3.49 3.26 3.89 3.36 3.76   NA 3.09 2.88 4.20 3.74 3.17 4.00
## [31] 3.78 3.40 2.78 2.81 4.30 3.16 3.00 3.44 2.90 2.84 3.70 3.10 3.05 3.30 3.32
## [46] 3.06 3.10 3.65 3.60 3.50 2.95 3.60 3.67 3.23 3.43 3.20 3.54 2.68 3.87 3.25
## [61] 3.50 3.35 3.60 2.56
```

CALC

Il calcio, il minerale più presente all'interno del nostro organismo è essenziale per lo sviluppo e per la salute delle ossa. Non avendo valori -1 che stanno ad indicare l'informazione mancante, procediamo a sostituire i punti alle virgole

```
SOFA_1$CALC
```

```
##  [1] "9,9"                "9.6"                "9"
##  [4] "7.3"                "9,2"                "9,7"
##  [7] "9,6"                "8,6"                "9.1"
## [10] "7.2"                "9,5"                "9.6"
## [13] "8.5"                "8,8"                "10.1"
## [16] "9.1999999999999993" "9.9"                "9,4"
## [19] "8,8"                "8.6"                "9,4"
## [22] "9.1"                "880"                "9,1"
## [25] "8.8000000000000007" "9"                  "9,9"
## [28] "8,7"                "9"                  "9,7"
## [31] "9,5"                "8.3000000000000007" "8,4"
## [34] "8,7"                "10"                 "9.6"
## [37] "9"                  "9.6999999999999993" "8.5"
## [40] "10"                 "9,0"                "8.5"
## [43] "9,3"                "9,1"                "8,8"
## [46] "9"                  "9,7"                "9.6999999999999993"
## [49] "9,4"                "9"                  "9.3000000000000007"
## [52] "9.3000000000000007" "8.6999999999999993" "8.6"
## [55] "9"                  "8,8"                "8.8000000000000007"
## [58] "8.5"                "9,5"                "8,8"
## [61] "8.5"                "8.6999999999999993" "8.8000000000000007"
## [64] "8.1999999999999993"
```

```
SOFA_1$CALC = str_replace(SOFA_1$CALC, ",", ".")
SOFA_1$CALC = as.numeric(SOFA_1$CALC)
min(SOFA_1$CALC)
```

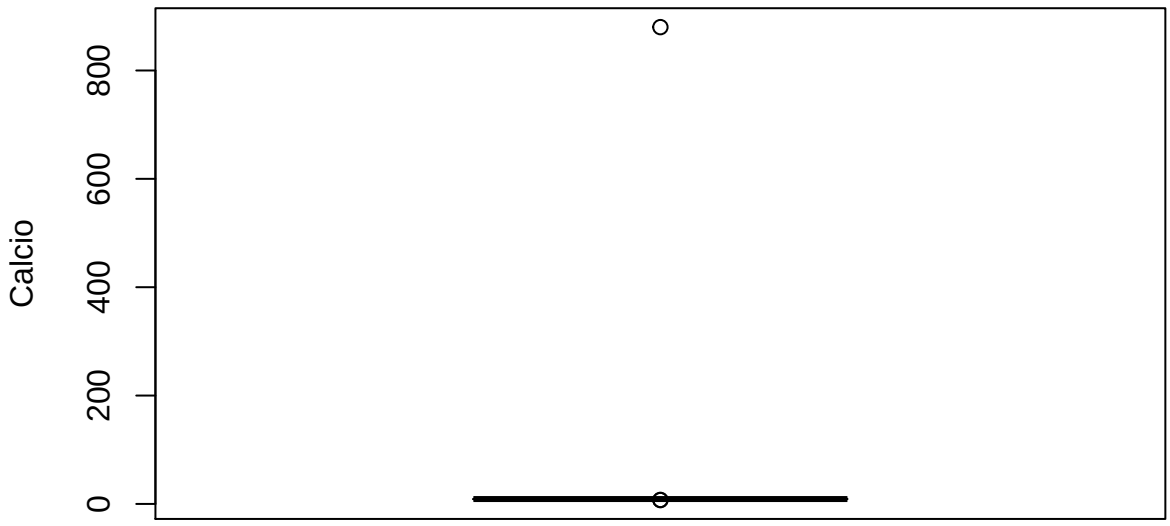
```
## [1] 7.2
```

```
max(SOFA_1$CALC)
```

```
## [1] 880
```

Per la variabile calcio notiamo un valore max molto alto. Probabilmente vi è stato un errore nella registrazione del dato.

```
p = boxplot(SOFA_1$CALC, ylab = 'Calcio')
```

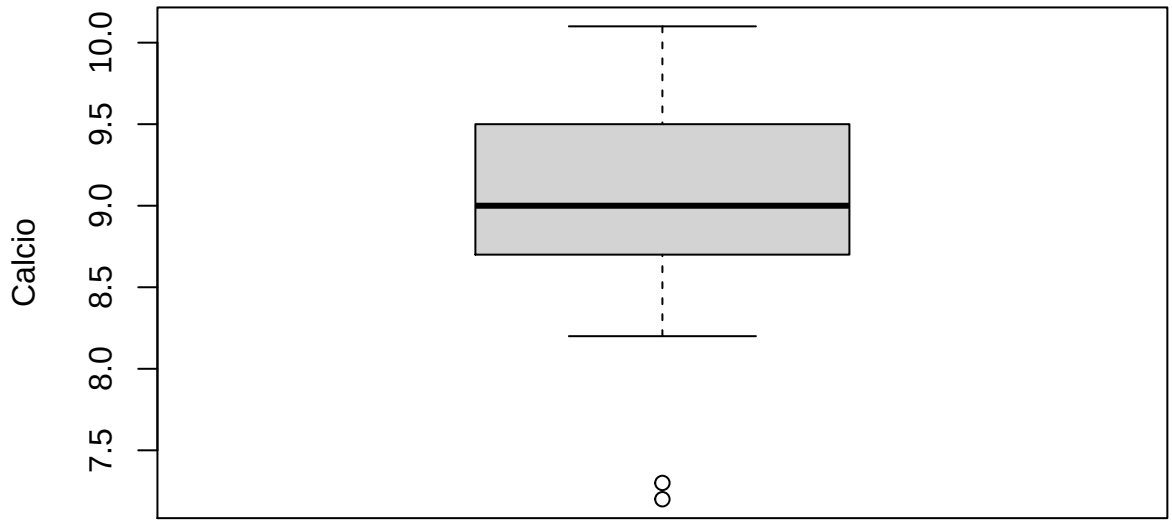



```
boxplot.stats(SOFA_1$CALC)$out
```

```
## [1]   7.3   7.2 880.0
```

Osservando il boxplot infatti il valore 880 è evidentemente un outlier. Mantenendo sempre l'idea di un errore nella trascrizione del dato si pensa che 880 si riferisca a un valore di calcio di 8.8 molto più coerente.

```
SOFA_1$CALC[23]= SOFA_1$CALC[23]/100

p = boxplot(SOFA_1$CALC, ylab = 'Calcio')
```




```
boxplot.stats(SOFA_1$CALC)$out
```

```
## [1] 7.3 7.2
```

VITD

La variabile VITD si riferisce ai livelli di vitamina D. Guardano i dati si osserva la presenza di valori -1 riferiti ai dati mancanti. Sostituiamo quindi il -1 con NA e le virgole con i punti permettendo così la classificazione della varibile come numerica.

```
SOFA_1$VITD
```

```
##  [1] "-1"              "3.1"             "9.1"
##  [4] "3"               "58"              "-1"
##  [7] "11,7"            "5,7"             "-1"
## [10] "3"               "-1"              "-1"
## [13] "-1"              "21,5"            "3"
## [16] "14.6"            "7.8"             "26"
## [19] "3,2"             "3.1"             "3,0"
## [22] "3"               "3"               "3"
## [25] "3"               "3"               "-1"
## [28] "3,3"             "3"               "9,3"
## [31] "18,4"            "-1"              "3"
## [34] "3"               "3,2"             "21.6"
```

```
## [37] "4.9000000000000004" "25.2"               "3"
## [40] "-1"                 "11"                "-1"
## [43] "8,3"                "3,7"               "22,1"
## [46] "4.9000000000000004" "-1"                "58.7"
## [49] "-1"                 "4.9000000000000004" "3"
## [52] "-1"                 "3.1"               "7.7"
## [55] "6"                  "21,8"              "6.6"
## [58] "-1"                 "20,1"              "-1"
## [61] "5.0999999999999996" "10.3"              "-1"
## [64] "67.099999999999994"
```

```
SOFA_1$VITD = str_replace(SOFA_1$VITD, ",", ".")
SOFA_1$VITD = ifelse(SOFA_1$VITD==-1,NA,SOFA_1$VITD)

SOFA_1$VITD = as.numeric(SOFA_1$VITD)
min(na.omit(SOFA_1$VITD))
```

```
## [1] 3
```

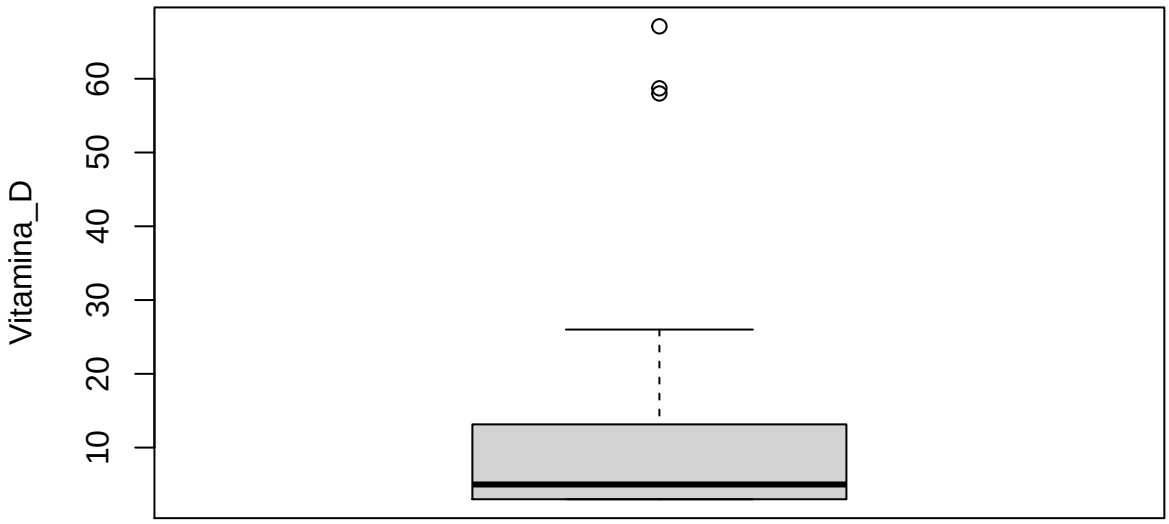
```
max(na.omit(SOFA_1$VITD))
```

```
## [1] 67.1
```

```
SOFA_1$VITD
```

```
##  [1]   NA  3.1  9.1  3.0 58.0   NA 11.7  5.7   NA  3.0   NA   NA   NA 21.5  3.0
## [16] 14.6  7.8 26.0  3.2  3.1  3.0  3.0  3.0  3.0  3.0  3.0   NA  3.3  3.0  9.3
## [31] 18.4   NA  3.0  3.0  3.2 21.6  4.9 25.2  3.0   NA 11.0   NA  8.3  3.7 22.1
## [46]  4.9   NA 58.7   NA  4.9  3.0   NA  3.1  7.7  6.0 21.8  6.6   NA 20.1   NA
## [61]  5.1 10.3   NA 67.1
```

```
p = boxplot(SOFA_1$VITD, ylab = 'Vitamina_D')
```




```
boxplot.stats(SOFA_1$VITD)$out
```

```
## [1] 58.0 58.7 67.1
```

Dal boxplot non si evidenziano grosse problematiche a livello di dati, perciò possiamo terminare così la nostra pulizia della variabile.

HBING

HBING è la variabile riferita all'emoglobina, una proteina che si trova all'interno dei globuli rossi. Analizzando i dati non troviamo valori mancanti o incongruenti. La modifica che possiamo apportare è quella del cambio virgola-punto per permettere la classificazione numerica della variabile.

```
SOFA_1$HBING
```

```
##  [1] "13,2"              "14.1"              "13"
##  [4] "13.3"              "11,8"              "14,3"
##  [7] "12,3"              "12,7"              "13"
## [10] "11.3"              "13,4"              "12.7"
## [13] "12.9"              "12,2"              "13.3"
## [16] "13"                "11.6"              "12,8"
## [19] "14,6"              "12.4"              "12,7"
## [22] "13.3"              "12.2"              "16,2"
## [25] "14.3"              "12.2"              "13,2"
```

```
## [28] "14,1"               "10.1"               "12,4"
## [31] "13,2"               "10.9"               "14,1"
## [34] "12,2"               "10,3"               "14.3"
## [37] "8.8000000000000007" "10.1"               "12.5"
## [40] "12.1"               "12,1"               "10.199999999999999"
## [43] "10,9"               "13,5"               "12,8"
## [46] "10.4"               "13,4"               "12.7"
## [49] "13,4"               "12.9"               "11.8"
## [52] "13.7"               "12.9"               "9.6"
## [55] "12.8"               "14,3"               "12.2"
## [58] "11.2"               "14,6"               "9,8"
## [61] "12.9"               "13.8"               "11.1"
## [64] "7.4"
```

```
SOFA_1$HBING = str_replace(SOFA_1$HBING, ",", ".")
SOFA_1$HBING = as.numeric(SOFA_1$HBING)
min(na.omit(SOFA_1$HBING))
```

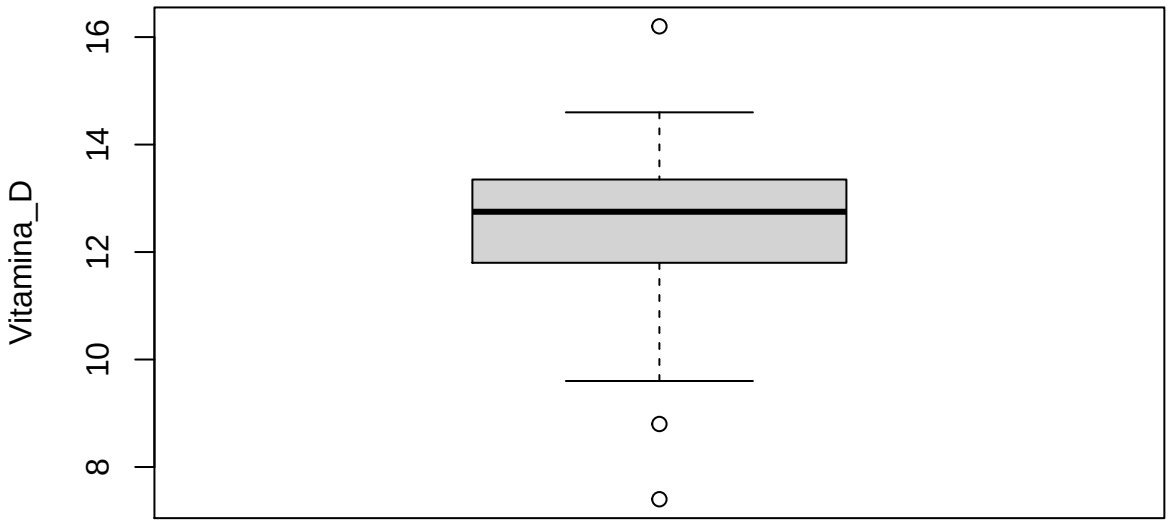
```
## [1] 7.4
```

```
max(na.omit(SOFA_1$HBING))
```

```
## [1] 16.2
```

boxplot emoglobina

```
p = boxplot(SOFA_1$HBING, ylab = 'Vitamina_D')
```




```
boxplot.stats(SOFA_1$HBING)$out
```

```
## [1] 16.2  8.8  7.4
```

TEMPRIC

TEMPRIC si riferisce al tempo di ricovero di un paziente in minuti. Nella variabile si evidenziano dei -1 riferiti al valore non disponibile e -2 dato che indica la provenienza del paziente da un altro ospedale. Ai fini dell'analisi questi due valori alterano i nostri risultati. Quello che si può fare è indicarli come NA.

```
s1 = SOFA_1
s1$TEMPRIC = ifelse(s1$TEMPRIC<0,NA,s1$TEMPRIC)
min(na.omit(s1$TEMPRIC))
```

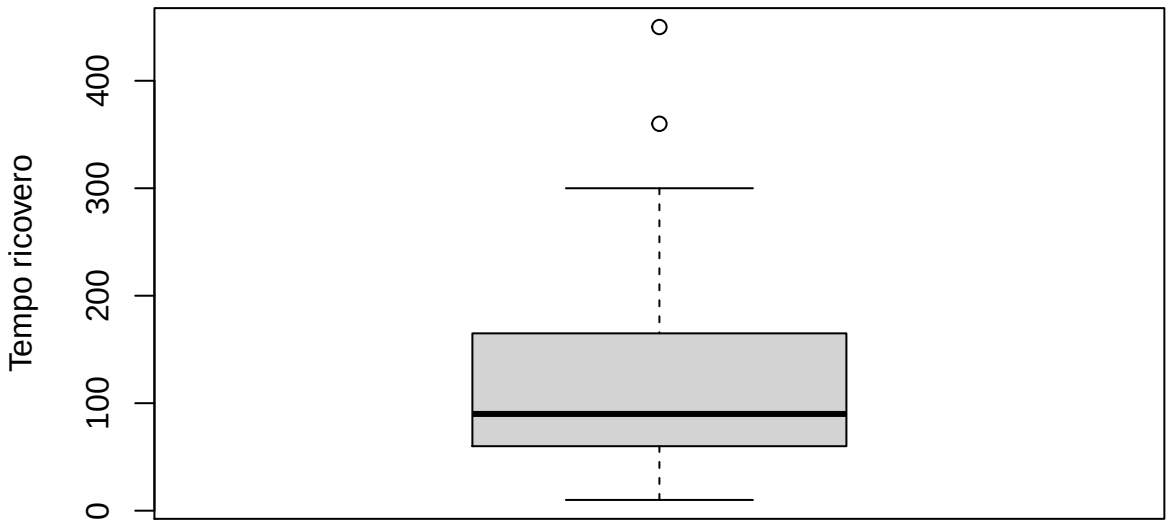
```
## [1] 10
```

```
max(na.omit(s1$TEMPRIC))
```

```
## [1] 450
```

boxplot tempo ricovero

```
bp_tempo = boxplot(s1$TEMPRIC, ylab = 'Tempo ricovero')
```




```
boxplot.stats(s1$TEMPRIC)$out
```

```
## [1] 450 360
```

DATDIM

La variabile DATDIM è il riferimento alla data di dimissione del paziente. Come possiamo osservare non ci sono date strutturalmente inadatte. Anche la classe della variabile è già in formato data.

```
s1$DATDIM
```

```
##  [1] "2011-11-02 UTC" NA               "2011-12-28 UTC" NA
##  [5] "2011-12-01 UTC" "2011-12-09 UTC" "2011-12-16 UTC" "2011-11-04 UTC"
##  [9] "2012-02-29 UTC" NA               "2011-10-24 UTC" "2012-01-10 UTC"
## [13] "2012-06-18 UTC" "2011-11-29 UTC" "2012-06-22 UTC" NA
## [17] "2012-01-05 UTC" "2011-11-10 UTC" "2012-03-09 UTC" NA
## [21] "2011-11-16 UTC" "2012-02-29 UTC" NA               "2012-03-09 UTC"
## [25] NA               "2012-02-24 UTC" "2011-12-12 UTC" "2011-10-14 UTC"
## [29] "2011-12-27 UTC" "2011-11-28 UTC" "2011-11-16 UTC" "2011-12-19 UTC"
## [33] NA               "2012-03-08 UTC" "2011-12-16 UTC" "2012-02-23 UTC"
## [37] NA               "2012-01-03 UTC" "2012-02-23 UTC" "2012-01-10 UTC"
```

```
## [41] "2011-11-14 UTC" "2012-06-19 UTC" "2011-11-18 UTC" "2011-10-14 UTC"
## [45] "2011-11-07 UTC" "2012-06-13 UTC" "2011-10-28 UTC" "2012-02-17 UTC"
## [49] "2011-10-28 UTC" "2012-01-12 UTC" "2012-06-19 UTC" "2012-02-15 UTC"
## [53] "2011-12-05 UTC" "2012-02-08 UTC" "2012-06-13 UTC" "2011-11-11 UTC"
## [57] "2012-02-29 UTC" "2012-03-02 UTC" "2012-02-16 UTC" NA
## [61] NA                "2011-12-28 UTC" "2011-12-21 UTC" NA
```

```
class(s1$DATDIM)
```

```
## [1] "POSIXct" "POSIXt"
```

DATAINT

Stesse osservazioni le possiamo fare per DATAINT che indica la data dell'intervento a cui tutti i pazienti sono stati sottoposti. La classe della variabile formato data.

```
s1$DATINT
```

```
##  [1] "2011-10-19 UTC" "2012-01-13 UTC" "2011-12-12 UTC" "2012-01-12 UTC"
##  [5] "2011-11-21 UTC" "2011-12-02 UTC" "2011-11-18 UTC" "2011-10-14 UTC"
##  [9] "2012-02-20 UTC" "2012-01-16 UTC" "2011-10-12 UTC" "2011-12-29 UTC"
## [13] "2015-06-12 UTC" "2011-11-15 UTC" "2012-06-15 UTC" "2012-01-12 UTC"
## [17] "2011-12-29 UTC" "2011-10-31 UTC" "2012-02-28 UTC" "2012-01-23 UTC"
## [21] "2011-11-09 UTC" "2012-02-23 UTC" "2012-01-30 UTC" "2012-02-27 UTC"
## [25] "2012-01-09 UTC" "2012-02-15 UTC" "2011-12-05 UTC" "2011-10-05 UTC"
## [29] "2011-12-27 UTC" "2011-11-18 UTC" "2011-11-07 UTC" "2011-12-09 UTC"
## [33] "2012-02-21 UTC" "2012-03-02 UTC" "2011-12-05 UTC" "2012-02-12 UTC"
## [37] "2012-01-16 UTC" "2011-12-22 UTC" "2012-02-15 UTC" "2011-12-27 UTC"
## [41] "2011-11-04 UTC" "2012-06-14 UTC" "2011-11-11 UTC" "2011-10-03 UTC"
## [45] "2011-10-26 UTC" "2012-06-01 UTC" "2011-10-12 UTC" "2012-02-10 UTC"
## [49] "2011-10-24 UTC" "2011-12-22 UTC" "2012-06-14 UTC" "2012-02-02 UTC"
## [53] "2011-11-23 UTC" "2012-02-01 UTC" "2012-06-08 UTC" "2011-11-24 UTC"
## [57] "2012-02-21 UTC" "2012-02-23 UTC" "2012-02-07 UTC" "2012-01-16 UTC"
## [61] "2012-01-30 UTC" "2011-12-16 UTC" "2011-12-14 UTC" "2012-01-23 UTC"
```

```
class(s1$DATINT)
```

```
## [1] "POSIXct" "POSIXt"
```

Quello che si può notare però sono delle incongruenze nelle date di dimissione e di intervento. L'idea è stata quella di formare un nuovo dataset rinominandolo s4 in cui sono inseriti solo i soggetti la cui data di intervento era prima della data di dimissione. Il contrario ovviamente avrebbe portato a delle distorsioni.

```
s4 = s1
(table(ifelse((difftime(s1$DATDIM, s1$DATINT, units = "days")>=0),TRUE,FALSE)))
```

```
##
## FALSE  TRUE
##     2    50
```

```
s4 <- subset(s1, (difftime(s1$DATDIM, s1$DATINT, units = "days")>=0)==TRUE | is.na(s1$DATDIM))
```

Dalla funzione table risultano 2 pazienti in cui la data di dimissione è precedente a quella dell'intevento. Nel dataset s4 questi verranno esclusi. Si passa perciò da un dataset con 64 osservazioni a uno con 62.

ANEST

Indica la **tipologia di anestesia** effettuata al paziente, è una variabile che abbiamo ritenuto opportuno convertirla in fattoriale. Abbiamo inoltre creato una nuova variabile da aggiungere al dataset (ANEST_lev) in modo tale da avere un'informazione più specifica ed informativa della variabile.

```
s4$ANEST = as.factor(s4$ANEST)
s4$ANEST_lev[s4$ANEST== 1]='generale'
s4$ANEST_lev[s4$ANEST== 2]='spinale'
s4$ANEST_lev[s4$ANEST== 3]='peridurale'
s4$ANEST_lev[s4$ANEST== 4]='plessica'
s4$ANEST_lev[s4$ANEST== 5]='combinata'
s4$ANEST_lev[s4$ANEST== 6]='sedazione'
s4$ANEST_lev[s4$ANEST== 7]='locale assistita'
s4$ANEST_lev[s4$ANEST== 8]='altro'
```

DATA DECESSO

Data decesso presenta diverse ambiguità. Durante il caricamento del dataset come è successo per la variabile NASCITA, i valori vengono visualizzati come numeri. Sono presenti inoltre dei -1 e una frase 'SI é RIFIUTATA' che deve essere necessariamente eliminata. Dapprima quindi convertiamo questi dati con NA. Successivamente convertiamo in classe data i resanti utilizzando come data di origine il 1899-12-30.

```
s4$`DATA DECESSO`
```

```
##  [1] NA              "40927"         "41010"          "40947"
##  [5] NA              NA              NA               "41273"
##  [9] NA              NA              NA               "40949"
## [13] NA              "-1"            "40563"          NA
## [17] NA              NA              NA               NA
## [21] NA              NA              NA               NA
## [25] NA              NA              "SI è RIFIUTATA" "40904"
## [29] NA              NA              NA               NA
## [33] NA              NA              NA               NA
## [37] NA              NA              "40965"          NA
## [41] NA              "40194"         "40830"          NA
## [45] "-1"            NA              NA               NA
## [49] "40921"         "-1"            "41034"          NA
## [53] NA              NA              NA               NA
## [57] NA              "40926"         NA               NA
## [61] NA              "40945"
```

```
class(s4$`DATA DECESSO`)
```

```
## [1] "character"
```

```
s4$`DATA DECESSO` = ifelse(s4$`DATA DECESSO`==-1,NA,s4$`DATA DECESSO`)
s4$`DATA DECESSO` = ifelse(s4$`DATA DECESSO`=='SI è RIFIUTATA',NA,s4$`DATA DECESSO`)

s4$`DATA DECESSO` = as.numeric(s4$`DATA DECESSO`)
(s4$`DATA DECESSO` = as.Date(x = s4$`DATA DECESSO`,origin = "1899-12-30", ))
```

```
##  [1] NA           "2012-01-19" "2012-04-11" "2012-02-08" NA
##  [6] NA           NA           "2012-12-30" NA           NA
## [11] NA           "2012-02-10" NA           NA           "2011-01-20"
## [16] NA           NA           NA           NA           NA
## [21] NA           NA           NA           NA           NA
## [26] NA           NA           "2011-12-27" NA           NA
## [31] NA           NA           NA           NA           NA
## [36] NA           NA           NA           "2012-02-26" NA
## [41] NA           "2010-01-16" "2011-10-14" NA           NA
## [46] NA           NA           NA           "2012-01-13" NA
## [51] "2012-05-05" NA           NA           NA           NA
## [56] NA           NA           "2012-01-18" NA           NA
## [61] NA           "2012-02-06"
```

Exploratory Data Analysis

DATA FINE FU

Per un analisi più approfondita abbiamo deciso di creare una variabile che sarà utile successivamente, denominata fine_fu, che indica il periodo in cui seguiremo i nostri pazienti dalla data di intervento fino ad un instante temporale definito. In questa variabile verranno inserite delle date. Per i pazienti che hanno una data di decesso, la data di decesso sarà la fine del follow-up, per i restanti terremo in considerazione il 31-12-2021.

```
s4$fine_fu = ifelse(is.na(s4$`DATA DECESSO`),NA,s4$`DATA DECESSO`)
s4$fine_fu =as.numeric(s4$fine_fu)
s4$fine_fu =as.Date(s4$fine_fu,origin = "1970-01-01", )
s4$fine_fu = ifelse(is.na(s4$fine_fu),as.Date("2021/12/31"),s4$fine_fu)
s4$fine_fu =as.Date(s4$fine_fu,origin = "1970-01-01", )
```

Andremo poi a calcolare i giorni di follow-up, ovvero i giorni che un paziente rimane in osservazione all'interno dello studio. Costruiremo una nuova variabile tenendo in considerazione come data indice la data di intervento. I giorni di follow up saranno dati dalla differenza tra la fine del follow-up e la data indice.

```
s4$giorni_fu = as.numeric(difftime(s4$fine_fu, s4$DATINT, units = "days"))
```

Osserveremo che vi sono valori negativi, ovvero si nota come vi sia un incongruenza tra la data di intervento e la data decesso di due pazienti. I due risultano essere deceduti prima della data di intervento, cosa non possibile. Procediamo dunque a creare un nuovo dataset non comprendendo questi pazienti in quanto potrebbero sorgere problematiche nel momento dell'analisi dei dati.

```
s5 = s4
s5 <- subset(s4, giorni_fu>=0)
```

BMI

Avendo a disposizione i dati riguardanti il peso e l'altezza di ogni paziente, possiamo calcolare il **BMI (Body mass index) come peso/altezza^2**. Per avere una più chiara interpretazione della variabile, classifichiamo i pazienti in:

- Sottopeso –> BMI < 18.5
- Normopeso –> 18.5 <= BMI < 25
- Obeso –> BMI >= 25

```
s5$BMI<-(s5$PESO)/((s5$ALTEZ/100)^2)

s5$BMI_cla = ifelse(s5$BMI<18.5,'Sottopeso',ifelse(s5$BMI<25,'Normopeso','Obeso'))
```

ETA'

Calcoliamo, inoltre, una delle variabili descrittive molto importanti, ovvero l'età. In questo caso abbiamo deciso di calcolarla alla data di intervento. Anche per questo dato si è creata una variabile categorica in modo da classificare i valori in range che vanno da:

- <60
- 60-69
- 70-79
- 80+

Si è notato che l'età dei pazienti al momento dell'intervento era elevata, infatti le uniche categorie che vengono visualizzate sono 70-79 e 80+.

```
s5$eta = trunc((s5$NASCITA %--% s5$DATINT) / years(1))

s5$eta_cla =  ifelse(s5$eta<60,'<60',
                     ifelse(s5$eta<70,'60-69',
                            ifelse(s5$eta<80,'70-79','80+')))

table(s5$eta_cla)
```

```
##
## 70-79   80+
##    11    49
```

Tabelle descrittive

```
#Tabella descrittiva
label(s5$sex_lev)= "GENERE"
label(s5$BMI) = "BMI"
label(s5$eta) = "ETA'"
label(s5$eta_cla) = "CLASSE ETA'"
label(s5$BMI_cla) = "CLASSE BMI"
label(s5$CCSCORE_cla) = "CLASSE CCSCORE"
```

```r
pvalue <- function(x, ...) {
    # Construct vectors of data y, and groups (strata) g
    y <- unlist(x)
    g <- factor(rep(1:length(x), times=sapply(x, length)))
    if (is.numeric(y)) {
        # For numeric variables, perform a standard 2-sample t-test
        p <- t.test(y ~ g)$p.value
    } else {
        # For categorical variables, perform a chi-squared test of independence
        p <- chisq.test(table(y, g))$p.value
    }
    # Format the p-value, using an HTML entity for the less-than sign.
    # The initial empty string places the output on the line below the variable label.
    c("", sub("<", "&lt;", format.pval(p, digits=3, eps=0.001)))
}

table1(~ sex_lev + eta  + eta_cla + BMI + BMI_cla  + CCSCORE_cla  | sofa_class, data=s5, overall=F, ext
```

```
## Get nicer `table1` LaTeX output by simply installing the `kableExtra` package
```

	Sofa Alto	Sofa Basso	P-value
	(N=18)	(N=42)	
GENERE			
donna	13 (72.2%)	38 (90.5%)	0.156
uomo	5 (27.8%)	4 (9.5%)	
ETA'			
Mean (SD)	86.7 (6.32)	84.7 (6.72)	0.29
Median [Min, Max]	86.0 [74.0, 102]	85.0 [70.0, 100]	
CLASSE ETA'			
70-79	2 (11.1%)	9 (21.4%)	0.56
80+	16 (88.9%)	33 (78.6%)	
BMI			
Mean (SD)	24.4 (5.06)	23.3 (4.65)	0.447
Median [Min, Max]	23.4 [17.3, 35.6]	23.1 [12.9, 34.5]	
Missing	0 (0%)	3 (7.1%)	
CLASSE BMI			
Normopeso	11 (61.1%)	22 (52.4%)	0.945
Obeso	5 (27.8%)	12 (28.6%)	
Sottopeso	2 (11.1%)	5 (11.9%)	
Missing	0 (0%)	3 (7.1%)	
CLASSE CCSCORE			
ccscore alto	8 (44.4%)	10 (23.8%)	0.049
ccscore basso	2 (11.1%)	18 (42.9%)	
ccscore medio	8 (44.4%)	14 (33.3%)	

Nella TAB1 vediamo una descrizione delle variabili principali classificate in base al punteggio Sofa. Per le variabili categoriche vengono confrontate le numerosità entro gruppi e le percentuali, mentre per quelle numeriche si osserva media con la SD e mediana con il range di appartenenza. Come prima annotazione di può osservare la maggior presenza di pazienti con sofa basso. La coorte di pazienti è prevalentemente di

genere femminile, con un'età molto alta, in entrambe le categorie di sofa siamo sopra agli 80 anni e con un BMI medio sotto il 25 perciò normopeso. Osservando infine la variabile ccscore si nota come per la categoria sofa alto si definiscono pazienti con un comorbidity score medio/alto mentre nella categoria sofa basso si hanno il 43% di pazienti con ccscore basso. Da notare bene anche la colonna dei p-value che risultano non significativi al 5% per tutte le variabili tranne che per la classe ccscore.

```
table1(~ CCSCORE_cla | sofa_class*sex_lev, data=s5,overall=F, extra.col=list('P-value'=pvalue), caption=
```

```
## Get nicer `table1` LaTeX output by simply installing the `kableExtra` package
```

	donna	uomo	donna	uomo	P-value
	(N=13)	(N=5)	(N=38)	(N=4)	
CLASSE CCSCORE					
ccscore alto	7 (53.8%)	1 (20.0%)	9 (23.7%)	1 (25.0%)	0.051
ccscore basso	1 (7.7%)	1 (20.0%)	18 (47.4%)	0 (0%)	
ccscore medio	5 (38.5%)	3 (60.0%)	11 (28.9%)	3 (75.0%)	

Siamo poi andati ad approfondire la distribuzione delle categorie del ccscore in relazione a quelle di genere e punteggio sofa. (TAB2) La numerosità maggiore la si osserva nel gruppo delle donne con sofa basso che è caratterizzata da un ccscore basso, mentre per le donne con sofa alto si denota circa un 54% avente un ccscore alto. Per gli uomini invece, seppur essendo in numero molto inferiore, hanno prevalentemente in entrambi i livelli di sofa, un ccscore medio, rispettivamente 75% per sofa basso e 60% per sofa alto.

Missing Value

SI vuole mostrare ora per ogni variabile, la percentuale dei Missing Value:

```
pl = plot_missing(s5,
             group = list(Good = 0.1, Bad = 1),
              missing_only = TRUE,
              geom_label_args = list("size" = 3, "label.padding" = unit(0.1, "lines")))
```

```
ti = ggtitle("Missing Value ")
ad = theme(plot.title = element_text(hjust = 0.5))
leg = scale_fill_discrete(limits = c("bad", "ok", "good"))
```

```
pl+ti+ad +theme(panel.background = element_rect(fill = "white" ))
```

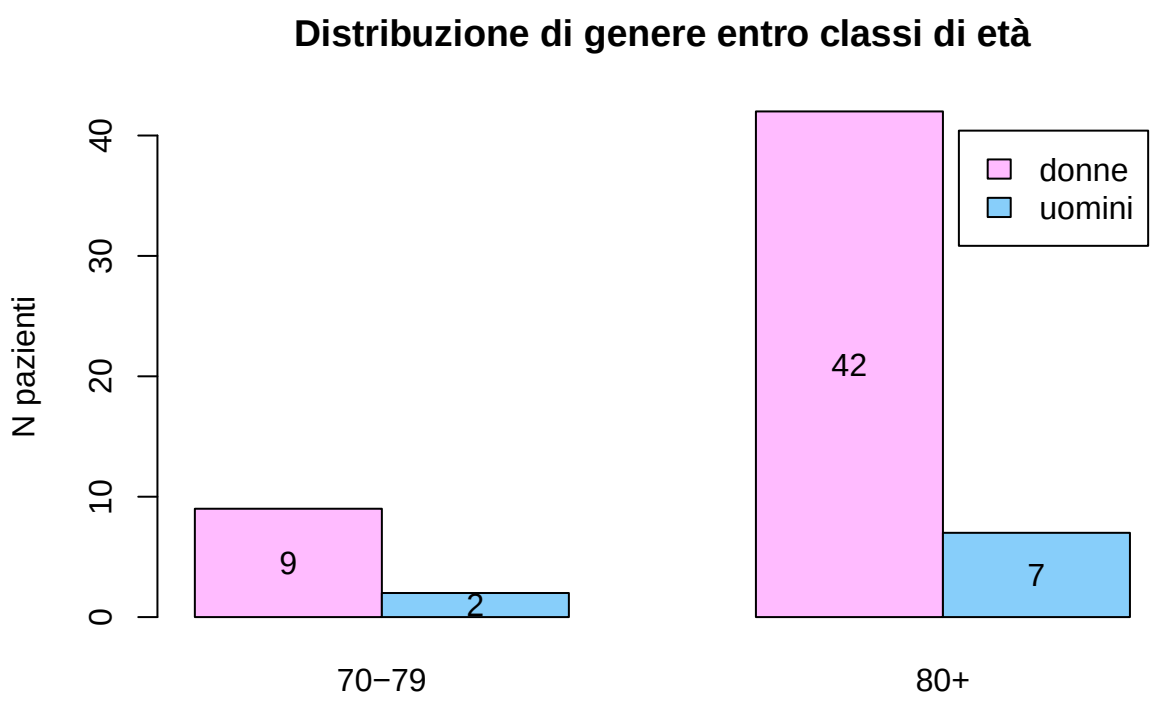
Questa osservazione la si può fare per individuare quali variabili, in fase di analisi, potranno portare a distorsioni nei risultati per dati mancanti. Ovviamente la data decesso è quella che contiene maggiori NA in quanto non tutti i pazienti sono deceduti. Infatti questa tecnica è più utile in variabili conteggio dove è necessario andare ad indagare più approfonditamente i risultati.

```
attach(s5)

(table1<-table(s5$sex_lev,s5$eta_cla))
```

Barplot classe età-genere

```
##
##         70-79 80+
##   donna     9  42
##   uomo      2   7
```

```
x1<-barplot((table1), col=c('plum1','lightskyblue'), beside=T,
        main="Distribuzione di genere entro classi di età",
        legend.text=c("donne","uomini"),
        names.arg=c('70-79', '80+'),ylab = 'N pazienti')
text(x=x1, y=table1/2, labels = table1)
```




```
attach(s5)
```

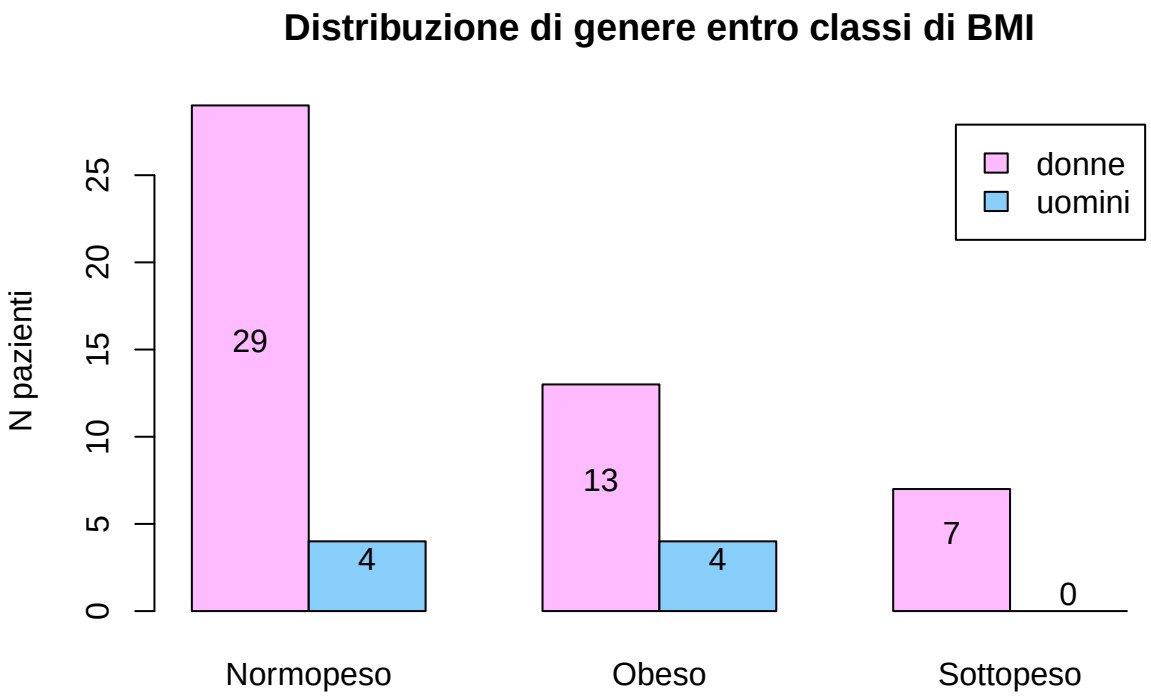
Barplot classe bmi-sesso

```
## The following objects are masked from s5 (pos = 3):
##
##     ALB, ALTEZ, ANEST, ANEST_lev, BMI, BMI_cla, CADUTE, CALC, CCSCORE,
##     CCSCORE_cla, DATA DECESSO, DATDIM, DATINT, eta, eta_cla, fine_fu,
##     giorni_fu, HBING, INTDURAT, MMSE, NASCITA, PAZIENTE, PESO, SEX,
##     sex_lev, sofa_class, SOFAING, STATCIV, STATCIV_lev, TEMPRIC, VITD
```

```
(table2<-table(s5$sex_lev,s5$BMI_cla))
```

```
##
##         Normopeso Obeso Sottopeso
##   donna        29    13         7
##   uomo          4     4         0
```

```
x2<-barplot((table2), col=c('plum1','lightskyblue'), beside=T,
        main="Distribuzione di genere entro classi di BMI",
        legend.text=c("donne","uomini"),
        names.arg=c('Normopeso', 'Obeso','Sottopeso'),ylab = 'N pazienti')
text(x=x2, y=(table2/2)+1, labels = table2)
```




```
attach(s5)
```

Barplot classe Sofa-sesso

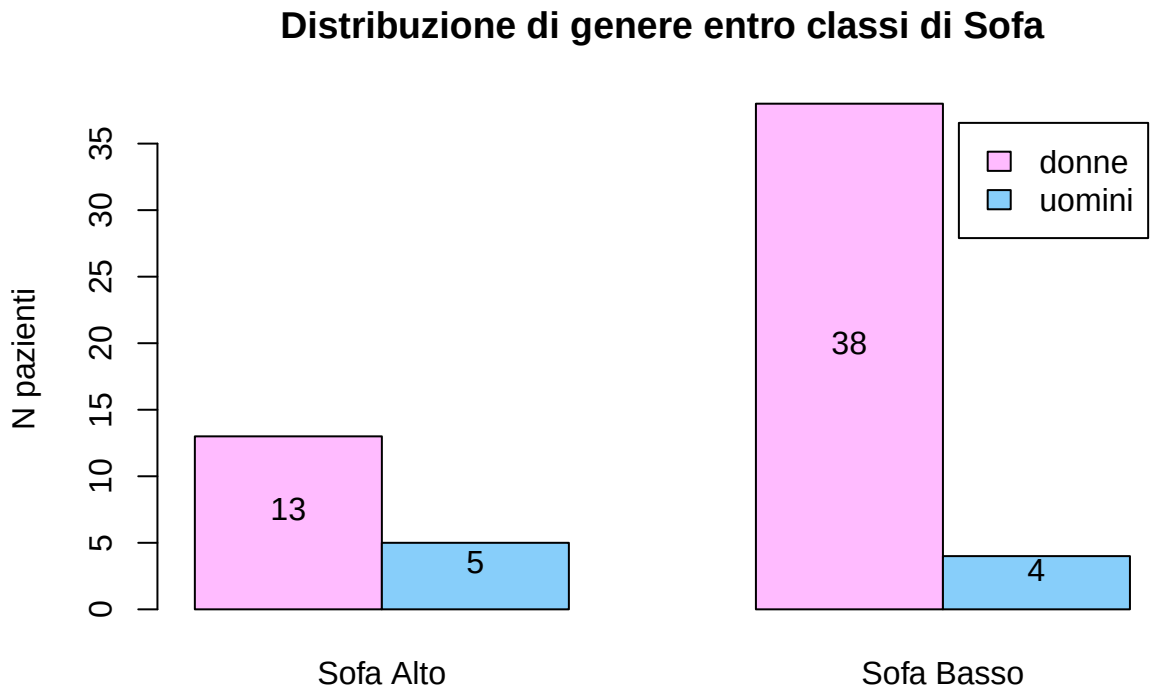
```
## The following objects are masked from s5 (pos = 3):
##
##     ALB, ALTEZ, ANEST, ANEST_lev, BMI, BMI_cla, CADUTE, CALC, CCSCORE,
##     CCSCORE_cla, DATA DECESSO, DATDIM, DATINT, eta, eta_cla, fine_fu,
##     giorni_fu, HBING, INTDURAT, MMSE, NASCITA, PAZIENTE, PESO, SEX,
##     sex_lev, sofa_class, SOFAING, STATCIV, STATCIV_lev, TEMPRIC, VITD

## The following objects are masked from s5 (pos = 4):
##
##     ALB, ALTEZ, ANEST, ANEST_lev, BMI, BMI_cla, CADUTE, CALC, CCSCORE,
##     CCSCORE_cla, DATA DECESSO, DATDIM, DATINT, eta, eta_cla, fine_fu,
##     giorni_fu, HBING, INTDURAT, MMSE, NASCITA, PAZIENTE, PESO, SEX,
##     sex_lev, sofa_class, SOFAING, STATCIV, STATCIV_lev, TEMPRIC, VITD
```

```
(table3<-table(s5$sex_lev,s5$sofa_class))
```

```
##
##         Sofa Alto Sofa Basso
```

```
##    donna         13          38
##    uomo           5           4
```

```
x3<-barplot((table3), col=c('plum1','lightskyblue'), beside=T,
        main="Distribuzione di genere entro classi di Sofa",
        legend.text=c("donne","uomini"),
        names.arg=c('Sofa Alto','Sofa Basso'),ylab = 'N pazienti')
text(x=x3, y=(table3/2)+1, labels = table3)
```

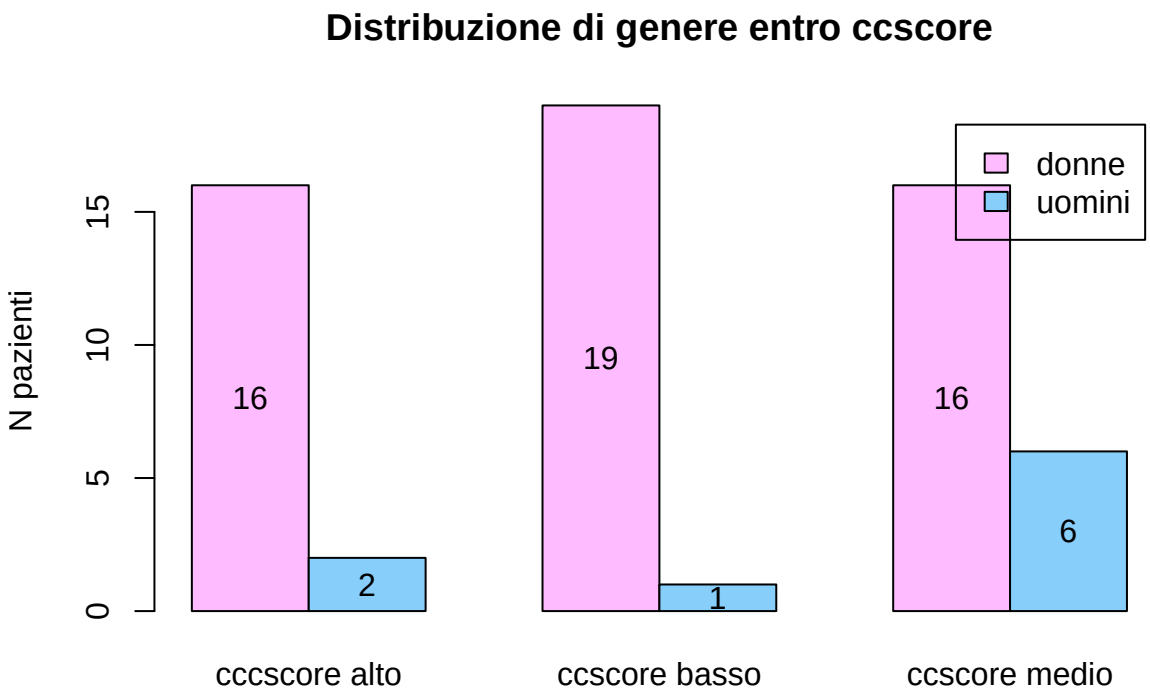



```
(table4<-table(s5$sex_lev, s5$CCSCORE_cla))
```

Barplot CCSCORE-sesso

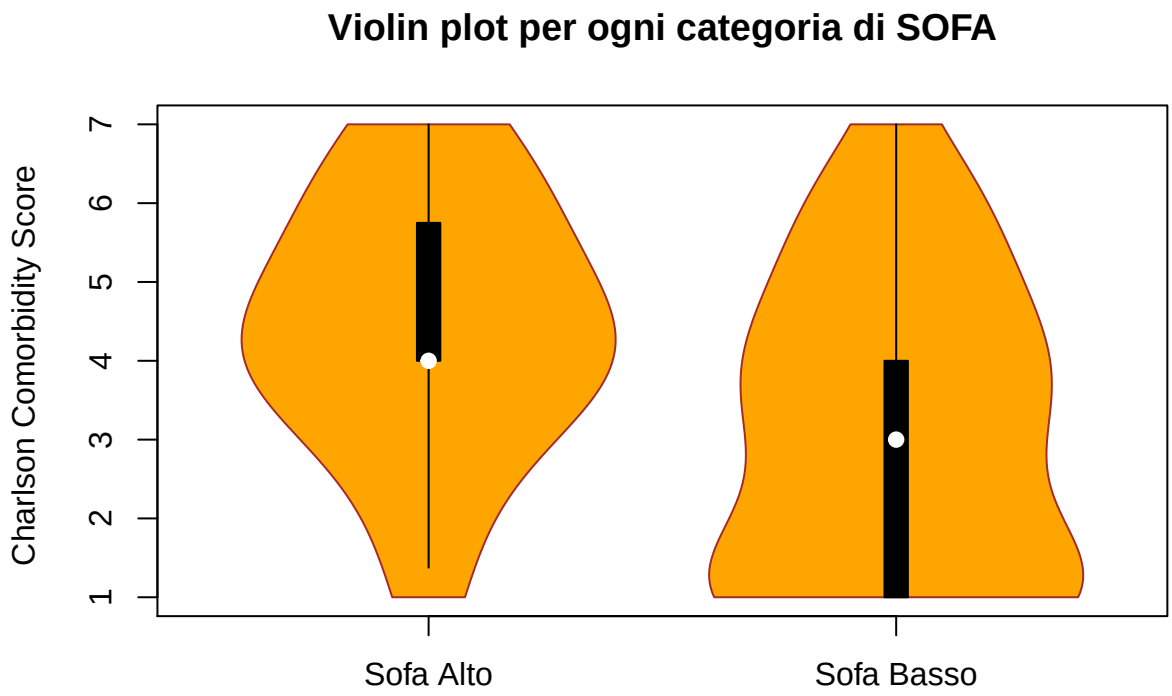
```
##
##         ccscore alto ccscore basso ccscore medio
##    donna           16            19            16
##    uomo             2             1             6
```

```
x4<-barplot((table4), col=c('plum1','lightskyblue'), beside=T,
        main="Distribuzione di genere entro ccscore",
        legend.text=c("donne","uomini"),
        names.arg=c('cccscore alto','ccscore basso','ccscore medio'),ylab = 'N pazienti')
text(x=x4, y=(table4/2), labels = table4)
```

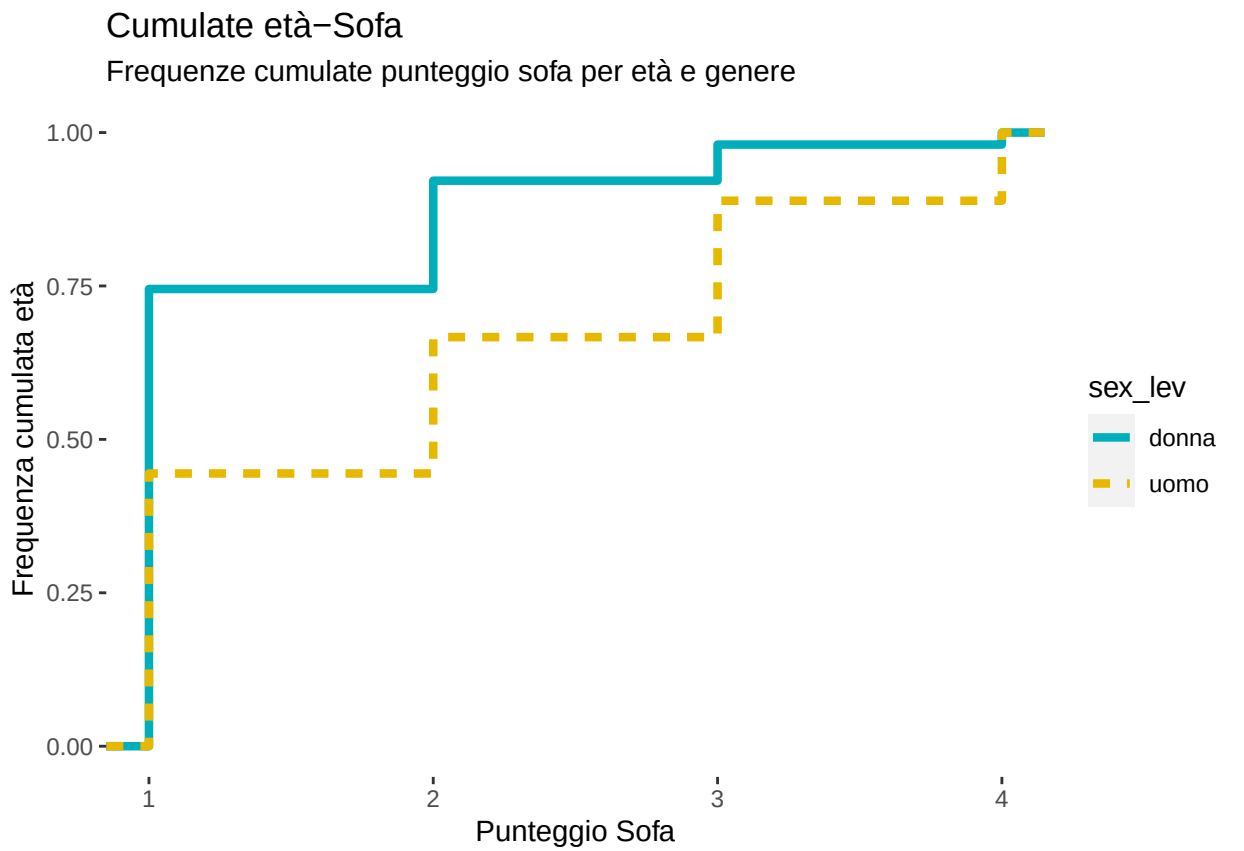


Violin plot di sofa per il punteggio ccscore

```
vioplot(s5$CCSCORE~s5$sofa_class,
data=s5,
method = "jitter",
main="Violin plot per ogni categoria di SOFA",
xlab = ' ',
ylab="Charlson Comorbidity Score ",
col="orange",
border="brown"
)
```



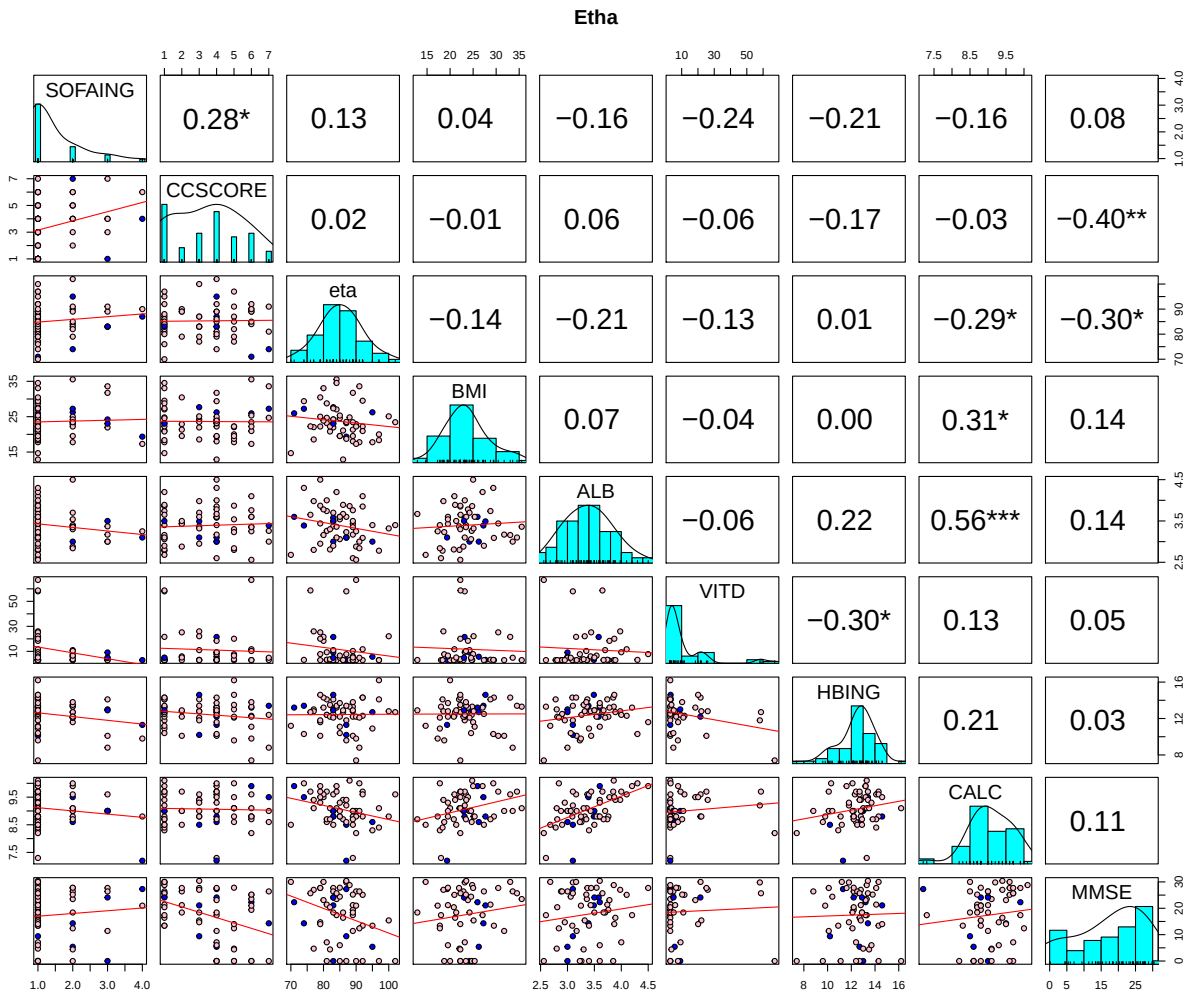

```
#funzione empirica cumulata dell'età divisa per sesso
sofaing_num = as.numeric(s5$SOFAING)
a <- ggplot(s5, aes(x = sofaing_num))+theme(panel.background = element_rect(fill = "white" ))+ ggtitle(

#funzione empirica cumulata
pl =a + stat_ecdf(aes(color = sex_lev,linetype = sex_lev),
              geom = "step", size = 1.5) +
  scale_color_manual(values = c("#00AFBB", "#E7B800"))+
  labs(y = "f(eta)")+theme(panel.background = element_rect(fill = "white" ))



bxp <- pl + labs(title = "Cumulate età-Sofa",
              subtitle = "Frequenze cumulate punteggio sofa per età e genere",
              x = "Punteggio Sofa", y = "Frequenza cumulata età")
bxp
```



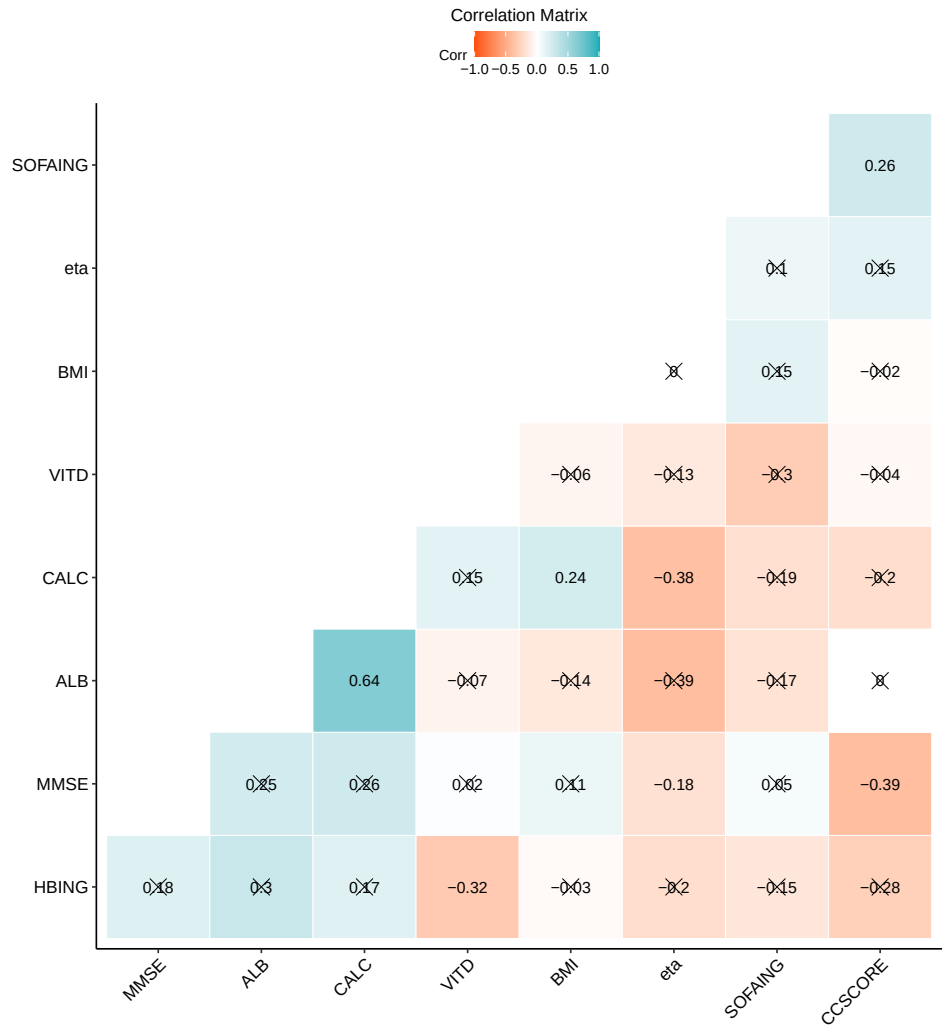
Matrice di correlazione

Un'analisi esplorativa molto importante è quella di verificare la correlazione tra le variabili. Questo è possibile grazie alla matrice di correlazione.

```
#da rivedere
s123<-s5[,c("SOFAING","CCSCORE","eta","BMI","ALB","VITD","HBING","CALC","MMSE")]# togliamo PESO e ALTEZ
#View(s123)

s123$SOFAING<-as.numeric(s123$SOFAING)
s123$CCSCORE<-as.numeric(s123$CCSCORE)
s123$eta<-as.numeric(s123$eta)
s123$BMI<-as.numeric(s123$BMI)

corr<-cor(na.omit(s123))

pairs.panels(s123, ellipses = F, lm=T,bg = c('blue','pink')[s5$SEX]
             ,pch= 21,  stars=TRUE, main = 'Etha')
```




```
ggcorrplot(na.omit(corr), p.mat = cor_pmat(s123), hc.order = TRUE,
           type = "lower",
           color = c("#FC4E07", "white", "#00AFBB"),
           outline.col = "white", lab = TRUE, ggtheme=custom_theme(), title = 'Correlation Matrix')
```



Osservando le matrici di correlazione, tenendo in considerazione solo le variabili maggiormente significative per questo tipo di analisi, si possono osservare correlazioni positive e negative. Il confronto più importante è, a nostro avviso, quello tra il punteggio sofa (SOFAING) e il CCSCORE. Le due variabili hanno un coefficiente di correlazione debolmente positivo (forse per la bassa numerosità del campione) ma significativo. QUesto definisce che per esempio, all'aumentare di una unità del ccscore, il sofaing aumenta di circa il 26%. Quindi se un paziente ha più malattie in concomitanza si può verificare un aumento del rischio di disfunzione di organi del 26%.

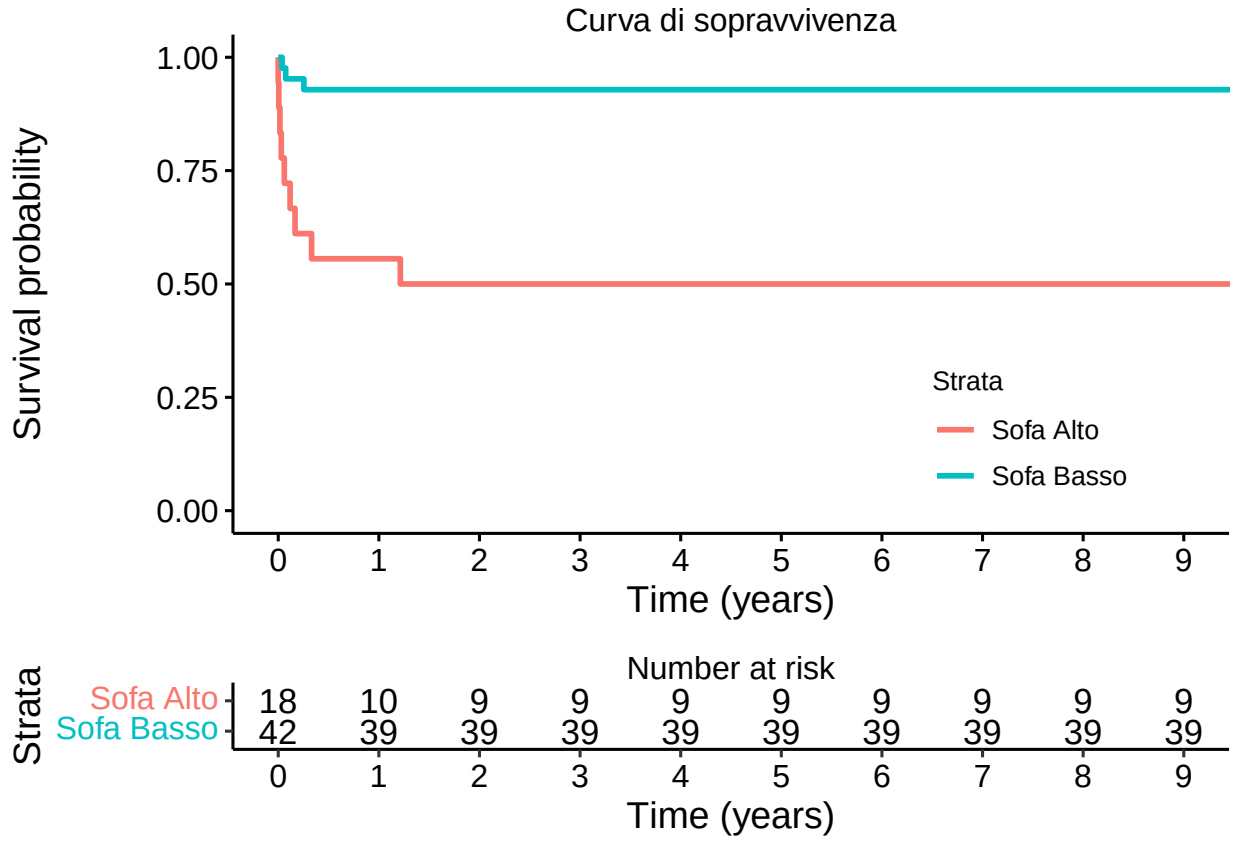
ANALISI SOPRAVVIVENZA

Una analisi possibile che si può effettuare con i dati a disposizione è l'analisi della sopravvivenza. Come primo passo andremo a creare una variabile 'EVENTO' in cui verrà assegnato 1 ai pazienti deceduti e 0 agli altri. Modificheremo anche il tempo del follow-up da giorni in anni.

Quello che cercheremo di fare è dimostrare che con un punteggio sofa alto, come mostrato in letteratura, aumenta il rischio di mortalità. La prima verifica che si può fare è tramite la curva di Kaplan-Meier.

```
s5$EVENTO<-ifelse(is.na(s5$`DATA DECESSO`), 0,1 )
#table(s5$EVENTO)
```

```r
s6 <- mutate(s5, all = EVENTO != "0")
#table(s6$all)

s6$anni_fu<- s6$giorni_fu/365
#View(s6)


fit_km <- survfit(Surv(anni_fu, all) ~ sofa_class, data = s6)

dat_km <- fortify(fit_km)

ggsurvplot(fit_km, title='Curva di sopravvivenza', ggtheme=custom_theme(),
           risk.table = TRUE, xlab = "Time (years)", censor = F, xlim = c(0,9),
           legend = c(0.8, 0.2),legend.labs = c('Sofa Alto','Sofa Basso'), break.x.by = 1)
```



Dal grafico si osserva una netta riduzione della curva riferita ai soggetti con sofa alto. Addirittura subito dopo un anno, il 50% dei pazienti aventi punteggio sofa alto all'inizio dello studio erano deceduti. Notiamo infatti 18 pazienti al tempo 0 e 10 rimasti in studio all'inizio del primo anno.Al contrario di quelli con sofa basso che hanno una riduzione di circa il 10%. Questa analisi, attraverso i dati disponibili, trova però delle difficoltà data la bassa numerosità di soggetti inclusi nello studio e il periodo di osservazione stabilito a priori tenendo in considerazione il periodo temporale in cui viene svolta questa relazione. Siamo fiduciosi però del fatto di osservare un risultato simile anche in database più numerosi.

MODELLO DI COX

Tenendo sempre in considerazione la bassa numerosità del dataset, successivamente alla curva di Kaplan-Meier è possibile definire il modello di Cox. Dapprima si determina il modello grezzo tenendo in considerazione solo la variabile di classificazione del sofa.

```
coxph2 <- coxph(Surv(giorni_fu,EVENTO)~sofa_class, data=s5)
summary(coxph2)
```

```
## Call:
## coxph(formula = Surv(giorni_fu, EVENTO) ~ sofa_class, data = s5)
##
##   n= 60, number of events= 12
##
##                          coef exp(coef) se(coef)      z Pr(>|z|)
## sofa_classSofa Basso -2.2137    0.1093   0.6681 -3.314 0.000921 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                      exp(coef) exp(-coef) lower .95 upper .95
## sofa_classSofa Basso    0.1093       9.15   0.02951    0.4048
##
## Concordance= 0.756  (se = 0.061 )
## Likelihood ratio test= 13.31  on 1 df,   p=3e-04
## Wald test            = 10.98  on 1 df,   p=9e-04
## Score (logrank) test = 16.17  on 1 df,   p=6e-05
```

Da questo primo modello si osserva come la variabile sofa basso sia molto significativa e abbia un effetto protettivo sulla mortalità rispetto alla variabile di riferimento sofa alto. Si osserva infatti un HR dello 0.109 con un intervallo di confidenza (0.029,0.405).

Considerando invece le altre covariate il modello diventa il seguente:

```
coxph1 <- coxph(Surv(giorni_fu,EVENTO)~sofa_class+eta_cla+BMI_cla+CCSCORE_cla, data=s5)
summary(coxph1)
```

```
## Call:
## coxph(formula = Surv(giorni_fu, EVENTO) ~ sofa_class + eta_cla +
##     BMI_cla + CCSCORE_cla, data = s5)
##
##   n= 57, number of events= 12
##    (3 observations deleted due to missingness)
##
##                                 coef  exp(coef)   se(coef)      z Pr(>|z|)
## sofa_classSofa Basso     -2.054e+00  1.282e-01  7.584e-01 -2.709  0.00676 **
## eta_cla80+                1.922e+01  2.229e+08  1.019e+04  0.002  0.99850
## BMI_claObeso             -3.358e-01  7.148e-01  8.146e-01 -0.412  0.68020
## BMI_claSottopeso          1.492e+00  4.447e+00  7.899e-01  1.889  0.05887 .
## CCSCORE_claccscore basso -7.285e-01  4.826e-01  1.220e+00 -0.597  0.55027
## CCSCORE_claccscore medio  7.885e-01  2.200e+00  7.087e-01  1.113  0.26591
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
##                           exp(coef) exp(-coef) lower .95 upper .95
## sofa_classSofa Basso     1.282e-01  7.801e+00   0.02899    0.5667
## eta_cla80+               2.229e+08  4.487e-09   0.00000       Inf
## BMI_claObeso             7.148e-01  1.399e+00   0.14482    3.5281
## BMI_claSottopeso         4.447e+00  2.249e-01   0.94559   20.9141
## CCSCORE_claccscore basso 4.826e-01  2.072e+00   0.04420    5.2691
## CCSCORE_claccscore medio 2.200e+00  4.545e-01   0.54850    8.8241
##
## Concordance= 0.863  (se = 0.042 )
## Likelihood ratio test= 23.26  on 6 df,   p=7e-04
## Wald test            = 12.94  on 6 df,   p=0.04
## Score (logrank) test = 22.82  on 6 df,   p=9e-04
```

La variabile sofa basso è ancora significativa. Il coefficiente passa da -2.2137 a -2.054 e l'HR da 0.109 a 0.128 con un IC (0.029,0.567). Si verifica perciò sempre l'effetto protettivo del sofa basso rispetto al sofa alto sulla mortalità come evidenziato anche dalla curva KM.