Progetto per Big Data in Health Care

Universitá di Milano-Bicocca

Matteo Corona - Lorenzo Lecce - Andrea Lucini Paioni

Contents

List of Figures

List of Tables

1 Introduzione

Questo progetto si propone di analizzare l'incidenza di seconda recidiva nei soggetti con resezione chirurgica del Carcinoma epatocellulare (HCC) che hanno già avuto una prima recidiva.
L'HCC è un tumore maligno del fegato e la resezione chirurgica è spesso utilizzata come opzione di trattamento. Tuttavia, anche dopo la rimozione del tumore, la recidiva è comune. Questo studio si concentrerà su pazienti che hanno già avuto una recidiva.
L'analisi dei dati sarà effettuata utilizzando metodi non parametrici e analisi univariate, come il modello Cox. L'obiettivo finale del progetto è quello di sviluppare un modello predittivo che possa aiutare a identificare i pazienti a rischio di seconda recidiva per poter adottare misure preventive appropriate.

1.1 Librerie

In prima istanza é necessario importare le librerie di *R* che verranno utilizzate in seguito.

```
# Importing libraries
library(kableExtra)
library(gridExtra)
library(ggplot2)
library(survival)
library(rms)
library(prodlim)
library(cmprsk)
library(skimr)
library(Greg)
library(splines)
library(corrplot)
library(pROC)
library(dcurves)
library(meta)
library(survminer)
library(riskRegression)
```

1.2 Importazione del dataset

Per importare il dataset nell'area di lavoro di *R* é sufficiente richiamare la funzione *read.csv()*. I dati che vengono importati non contengono valori nulli o errori e, in generale, non presentano problemi quindi non é necessario eseguire nessuna procedura di *preprocessing*.

```
# Reading .csv file
hcc = read.csv("HCC.csv", sep = "")
```

1.3 Il dataset

Il dataset che é stato fornito é composto da 10 differenti attributi. Di seguito sono riportate le prime osservazioni contenute nel dataset.

```
# Printing the first observations
kable(hcc[1:7,], booktabs = T, caption = "struttura del dataset") %>%
kable_styling(latex_options = c("striped", "scale_down", "HOLD_position"))
```

Table 1: struttura del dataset

idpat	Age	Gender	RecMultinodular	RecNoduleLargeSize	RecExtrahepatic	TimeToFirstRecMonths	FupAfterFirstRecMonths	SecondRecOrDeath	RecTreat
1	55	M	1	0	1	20.327869	2.038354	1	PAL
2	62	M	1	1	1	9.442623	9.665743	2	PAL
3	72	M	0	0	0	27.442623	51.649259	0	CUR
4	69	M	1	0	0	11.803279	20.252033	1	CUR
5	77	F	1	0	0	12.688525	87.912232	0	CUR
6	54	M	0	0	0	41.016393	56.613637	0	CUR
7	75	M	1	0	0	45.409836	26.958875	1	PAL

Gli attributi presenti nel dataset sono in seguenti:

- idpat: identificativo del paziente
- Age: età del paziente alla prima recidiva (in anni)
- Gender: genere del paziente (M o F)
- RecMultinodular: indicatore di prima recidiva multinodulare o a singolo nodulo (1 = multi; 0 = singolo)
- RecNoduleLargeSize: indicatore della dimensione del nodulo recidivante più grande (1 = se il nodulo è >5cm, 0 = altrimenti)
- RecExtrahepatic: indicatore di recidiva extraepatica (1 = extraepatica, 0 = solo epatica)
- TimeToFirstRecMonths: tempo tra la resezione del tuomore primario e la comparsa di recidiva (in mesi)
- FupAfterFirstRecMonths: tempo di follow-up dalla prima recidiva (in mesi)
- SecondRecOrDeath: indicatore di evento (1 = seconda recidiva, 2 = morte senza seconda recidiva, 0 = censura)
- RecTreat: trattamento della prima recidiva (CUR = altra resezione chirurgica o thermoablation, PAL = transarterial chemoembolization or Sorafenib)

2 Analisi descrittiva dei dati

Il primo passo per l'analisi dei dati é certamente quello di effettuare un'analisi descrittiva delle variabili che si hanno a disposizione.

2.1 Istogrammi variabili numeriche

In primo luogo è utile visualizzare degli istogrammi delle variabili numeriche del dataset. Il dataset fornito contiene le variabili numeriche *Age*, *TimeToFirstRecMonths* e *FupAfterFirstRecMonths*.

```
# Setting the canvas
par(mfrow=c(3,1))
# Histogram age
hist(hcc$Age, main = "Age", col = "red",
     xlim = c(30,90),
     ylim = c(0, 60),
     breaks = 24)
# Histogram TimeToFirstRecMonths
hist(hcc$TimeToFirstRecMonths, main = "TimeToFirstRecMonths", col = "red",
    breaks = 24,
    xlim = c(0,120),
    ylim = c(0, 80))
```

```r
# Histogram FupAfterFirstRecMonths
hist(hcc$FupAfterFirstRecMonths, main = "FupAfterFirstRecMonths", col = "red",
     xlim = c(0,100),
     ylim = c(0, 80),
     breaks = 20)
```

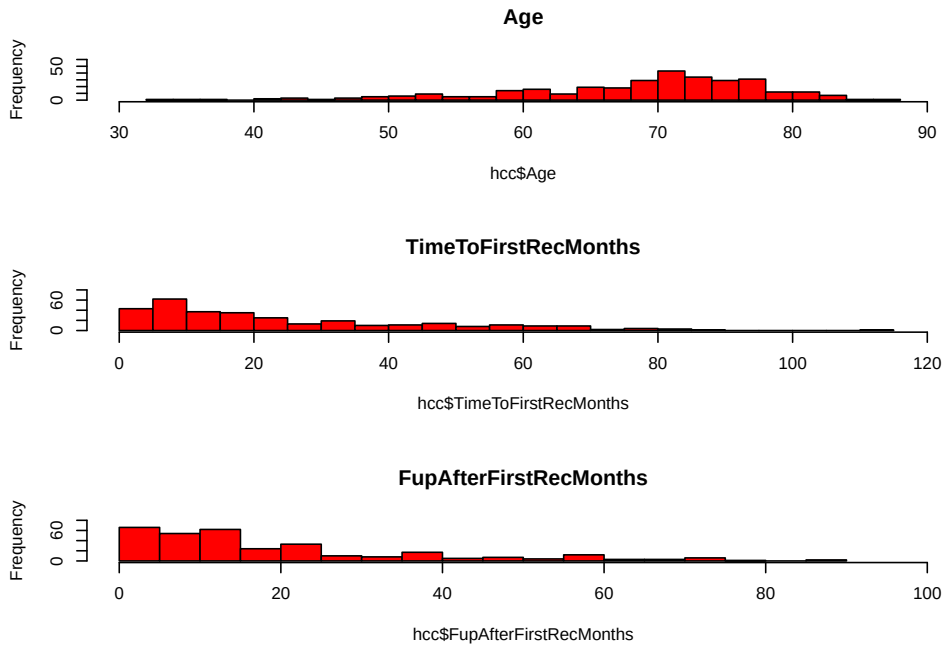

Figure 1: Istogrammi variabili numeriche

2.2 Correlazione

Successivamente è necessario guardare la correlazione tra le variabili numeriche e, in questo caso, l'ampiezza dei coefficienti di correlazione tra le variabili numeriche indica che esse non sono correlate tra loro in maniera significativa.

```r
# Calcolo matrice di correlazione
corrplot(cor(hcc[, c(2,6,7)]), method="circle", type = "full")
```

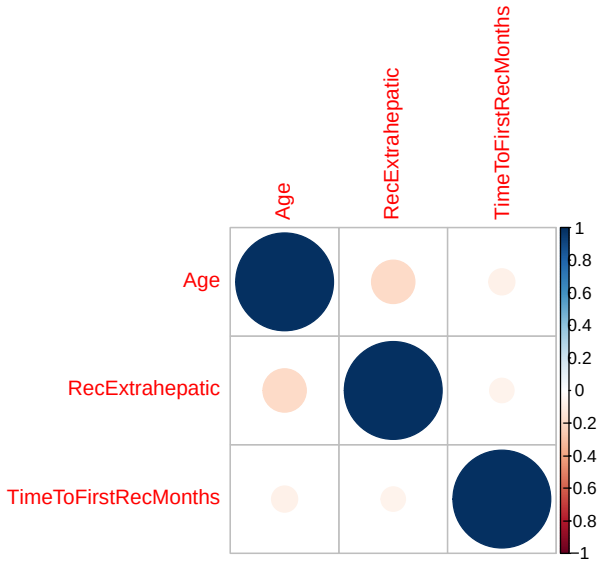

Figure 2: Matrice di correlazione

2.3 Variabili categoriche

A questo punto si procede con la descrizione delle variabili categoriche. Le variabili categoriche del dataset sono in tutto 6 e corrispondono a *Gender*, *RecMultinodular*, *RecNoduleLargeSize*, *RecExtrahepatic*, *SecondRecOrDeath*, e *RecTreat*. Per una prima analisi di queste variabili è sufficiente costruire dei *bar plot* che sono riportati di seguito.

```
# Setting the canvas
par(mfrow=c(2,3))
# Gender bar plot
mybar1 <- barplot(table(hcc$Gender), main = "Gender",
                  ylim = c(0, 300), col = c("red", "blue"))
# Graphic settings for bar plot 1
text(mybar1, table(hcc$Gender)+20 ,
     paste("n: ", table(hcc$Gender),
           sep="") ,cex=1)
# RecMultinodular bar plot
mybar2 <- barplot(table(hcc$RecMultinodular), main = "RecMultinodular",
                  ylim = c(0, 300), col = c("red", "blue"))
# Graphic settings for bar plot 2
text(mybar2, table(hcc$RecMultinodular)+20 ,
     paste("n: ", table(hcc$RecMultinodular),
           sep="") ,cex=1)
# RecNoduleLargeSize bar plot
mybar3 <- barplot(table(hcc$RecNoduleLargeSize), main = "RecNoduleLargeSize",
                  ylim = c(0, 300), col = c("red", "blue"))
# Graphic settings for bar plot 3
text(mybar3, table(hcc$RecNoduleLargeSize)+20 ,
     paste("n: ", table(hcc$RecNoduleLargeSize),
           sep="") ,cex=1)
# RecExtrahepatic bar plot
mybar4 <- barplot(table(hcc$RecExtrahepatic), main = "RecExtrahepatic",
                  ylim = c(0, 300), col = c("red", "blue"))
```

```
# Graphic settings for bar plot 4
text(mybar4, table(hcc$RecExtrahepatic)+20 ,
     paste("n: ", table(hcc$RecExtrahepatic),
           sep="") ,cex=1)
# SecondRecOrDeath bar plot
mybar5 <- barplot(table(hcc$SecondRecOrDeath), main = "SecondRecOrDeath",
                  ylim = c(0, 300), col = c("red", "blue", "green"))
# Graphic settings for bar plot 5
text(mybar5, table(hcc$SecondRecOrDeath)+20 ,
     paste("n: ", table(hcc$SecondRecOrDeath),
           sep="") ,cex=1)
# RecTreat bar plot
mybar6 <- barplot(table(hcc$RecTreat), main = "RecTreat",
                  ylim = c(0, 300), col = c("red", "blue"))
# Graphic settings for bar plot 6
text(mybar6, table(hcc$RecTreat)+20 ,
     paste("n: ", table(hcc$RecTreat),
           sep="") ,cex=1)
```

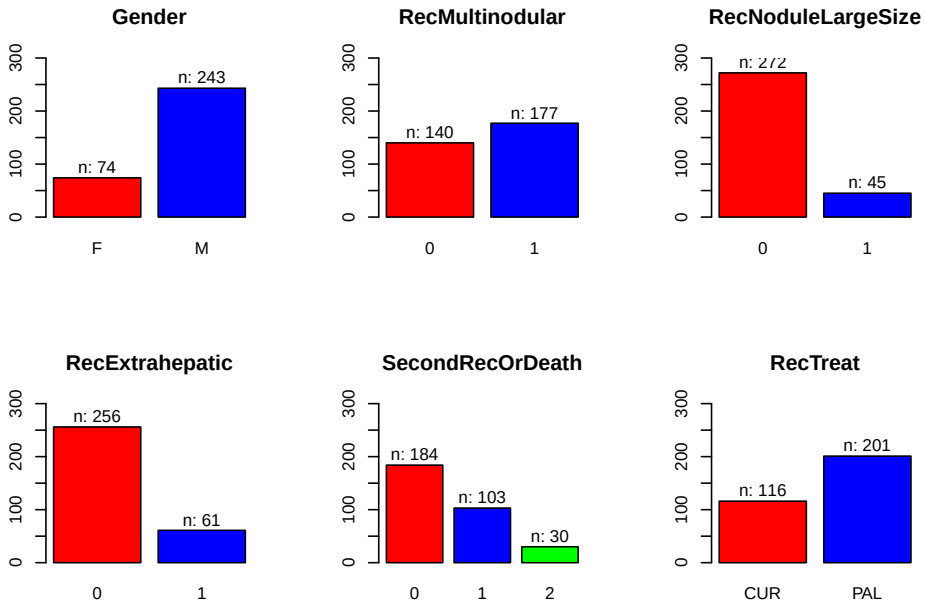

Figure 3: Categorical variables - bar plots

2.4 Confronto dei trattamenti

È possibile, ora, effettuare un primo confronto tra i due trattamenti costruendo opportunamente un istogramma che rappresenti la distribuzione dei conteggi di recidive a in funzione del tempo di recidiva, separando i due tipi di trattamento (CUR e PAL). Il risultato è il seguente grafico.

```
ggplot(hcc, aes(x = TimeToFirstRecMonths,
                fill = RecTreat,
                colour = RecTreat)) +
```

```
  geom_histogram(alpha = 0.5,
                 position = "identity")
```

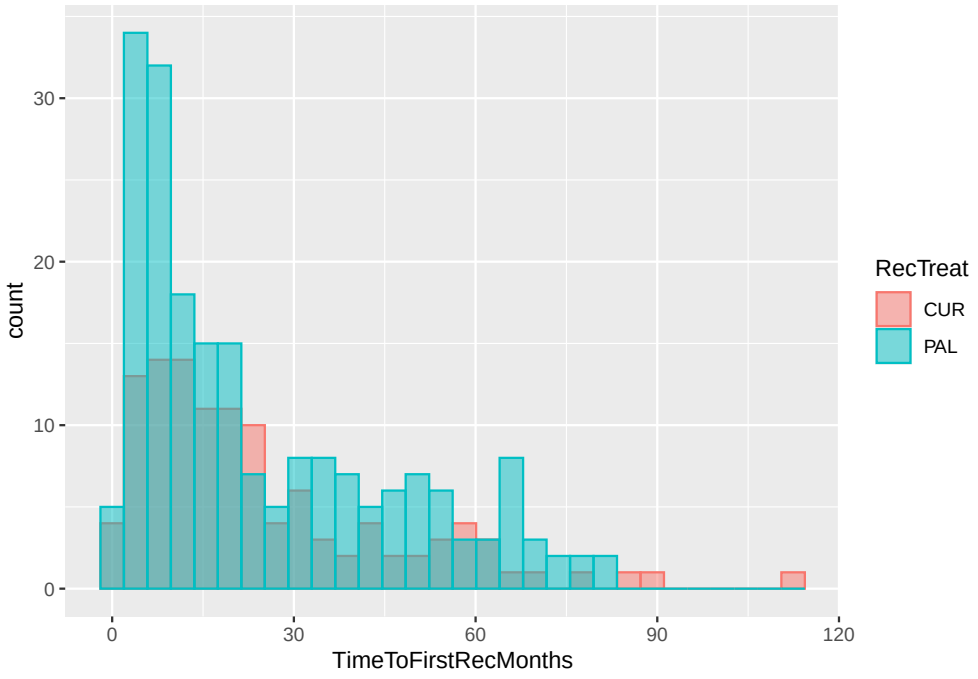

Figure 4: Threathment comparison

Osservando l'istogramma si può apprezzare anche solo visivamente che il trattamento *PAL* è associato ad un numero di conteggi maggiore.

2.5 Distribuzione del follow-up dopo la prima recidiva

In maniera analoga a quanto fatto per il confronto tra i due trattamenti, è possibile costruire una distribuzione del conteggio di di recidive in funzione del tempo evidenziando la differenza tra il genere maschile e quello femminile. In questo caso sì è optato per una distribuzione di densità, sfruttando il comando *geom_density()* della libreria *ggplot2*.

```
# Density plot del follow-up dopo la prima recidiva
ggplot(hcc, aes(x = FupAfterFirstRecMonths, fill = Gender)) +
  geom_density(alpha = 0.5) +
  scale_fill_manual(values = c("skyblue", "pink")) +
  labs(title = "Distribuzione del follow-up dopo la prima recidiva",
       x = "Tempo in mesi", y = "Densità")
```

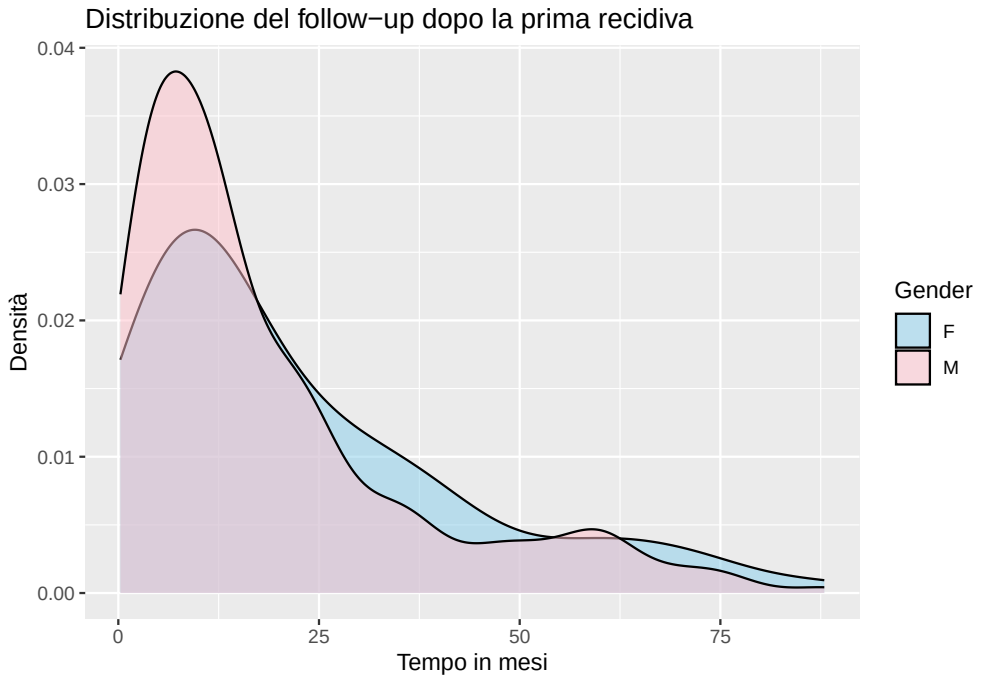

Figure 5: Distribuzione del follow-up dopo la prima recidiva

Da questa distribuzione si può apprezzare il fatto che c'è una differenza non trascurabile tra soggetti di sesso diverso (soprattutto nei primi 20 mesi)

3 Analisi non parametrica

Dopo aver effettuato una prima analisi descrittiva del dataset si può cominciare con effettuare delle analisi statistiche più dettagliate.

3.1 Curve di incidenza di Aalen-Johansen

Il seguente codice utilizza la funzione *prodlim()* per stimare la funzione di incidenza cumulativa e gli intervalli di confidenza per gruppo di trattamento per ogni evento di interesse (seconda recidiva e decesso senza seconda recidiva). Dopo aver stimato l' incidenza, la funzione *plot()* permette di generare opportunamente i due grafici che mostreranno la funzione di incidenza cumulativa nel tempo per ciascun gruppo di trattamento.

```
# Calling prodlim() function for creating incidence curves
crFit_sr <- prodlim(Hist(FupAfterFirstRecMonths,SecondRecOrDeath)~RecTreat, data=hcc)
# Setting the canvas
par(mar=c(4,2,3,1), mfrow=c(1,2))
# Plot first graph
plot(crFit_sr, cause=1, xlab="Time at event (months)", xlim=c(0,85), confint = T,
     legend.x="topleft", legend.legend=c("CUR","PAL"),
     atrisk = FALSE)
title(main = "Seconda recidiva")
# Plot second graph
plot(crFit_sr, cause=2, xlab="Time at event (months)", xlim=c(0,85), confint = T,
     legend.x="topleft", legend.legend=c("CUR","PAL"),
```

```
     atrisk = FALSE)
title(main = "Morte senza seconda recidiva")
```

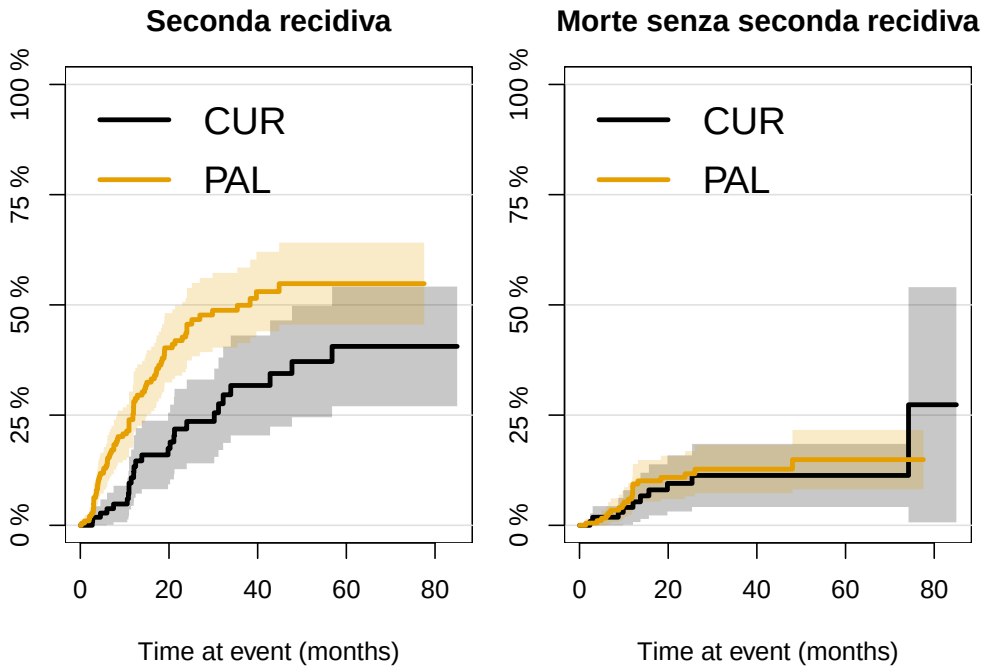

Figure 6: Curve di incidenza di Aalen-Johansen

3.2 Gray test

A questo punto, avendo calcolato le curve di incidenza, è possibile confrontarle sfruttanto il *Gray test*, che è implementato nelle seguenti righe di codice (viene sfruttata la libreria *kableExtra* per visualizzare in una tabella i risultati del test),

```
# Calcolo stime indicenza cumulativa
ci<-with(hcc, cuminc(FupAfterFirstRecMonths,SecondRecOrDeath,RecTreat))
# Printing the Gray test results
kable(round(ci$Tests, 3), booktabs = T, caption = "Gray test") %>%
kable_styling(latex_options = c("striped", "HOLD_position"))
```

Table 2: Gray test

stat	pv	df
10.284	0.001	1
0.223	0.637	1

La prima colonna "stat" corrisponde alla statistica del test. In questo caso, per il primo evento (seconda recidiva) la statistica del test è 10.284, mentre per il secondo evento (morte senza seconda recidiva) la statistica è 0.223.
La seconda colonna "pv" corrisponde al *p value* associato alla statistica del test. In questo caso, per il primo evento il *p value* è 0.001, mentre per il secondo evento il è 0.637.

Infine, l'ultima colonna "df" corrisponde ai gradi di libertà del test. In questo caso, per entrambi gli eventi i gradi di libertà sono 1, perchè ci sono solo due gruppi di confronto).
In sintesi, il test mostra una differenza significativa tra le funzioni di incidenza cumulativa dei due gruppi per il primo evento, ma non per il secondo evento. Questa differenza si può apprezzare anche graficamente in quanto nel primo grafico le curve ci incidenza si separano maggiormente rispetto a quanto avviene nel secondo grafico.

4 Modello di Cox

A seguito dell'analisi non parametrica svolta con le curve di Aalen-Johansen e con il test Gray, si può procedere con una analisi univariata.

4.1 Implementazione su *R* dei modelli

Il seguente codice *R* implementa il modello Cox, che permette di stimare l'associazione di ciascuna variabile indipendente con ciascuno dei due eventi competitivi (seconda recidiva - morte senza seconda recidiva) e con l'endpoint composito (seconda recidiva o morte).

```
# Creating Cox model for second rec
cox_model_rec <- coxph(Surv(FupAfterFirstRecMonths, SecondRecOrDeath==1) ~ Age + Gender +
                         RecMultinodular + RecNoduleLargeSize + RecExtrahepatic +
                         TimeToFirstRecMonths + RecTreat, hcc)
# Printing results using kable
result_rec <- finalfit::fit2df(cox_model_rec, condense = FALSE)
kable(result_rec, booktabs = T,
      caption = "modello di Cox per seconda recidiva") %>%
kable_styling(latex_options = c("striped", "HOLD_position"))
```

Table 3: modello di Cox per seconda recidiva

explanatory	HR	L95	U95	p
Age	1.0163930	0.9917178	1.0416822	0.1947271
GenderM	1.5627381	0.9491345	2.5730288	0.0792989
RecMultinodular	1.6915006	1.0798607	2.6495772	0.0217039
RecNoduleLargeSize	1.7509723	1.0381118	2.9533468	0.0357115
RecExtrahepatic	1.1325233	0.6857364	1.8704113	0.6268502
TimeToFirstRecMonths	0.9758013	0.9633352	0.9884286	0.0001883
RecTreatPAL	1.6871207	1.0314740	2.7595231	0.0372144

```
# Creating Cox model for death without second rec
cox_model_death <- coxph(Surv(FupAfterFirstRecMonths, SecondRecOrDeath==2) ~ Age + Gender +
                           RecMultinodular + RecNoduleLargeSize + RecExtrahepatic +
                           TimeToFirstRecMonths + RecTreat, hcc)
# Printing results using kable
result_death <- finalfit::fit2df(cox_model_death, condense = FALSE)
kable(result_death, booktabs = T,
      caption = "modello di Cox per morte senza seconda recidiva") %>%
kable_styling(latex_options = c("striped", "HOLD_position"))
```

Table 4: modello di Cox per morte senza seconda recidiva

explanatory	HR	L95	U95	p
Age	1.0157406	0.9738923	1.059387	0.4668782
GenderM	0.8854076	0.3972787	1.973291	0.7659684
RecMultinodular	1.2679008	0.5685631	2.827430	0.5618651
RecNoduleLargeSize	2.2379354	0.9102780	5.502006	0.0792345
RecExtrahepatic	1.5187318	0.6292372	3.665623	0.3526196
TimeToFirstRecMonths	0.9991399	0.9804004	1.018237	0.9290196
RecTreatPAL	1.2313122	0.5241823	2.892371	0.6329670

```
# Creating Cox model for composite endpoint
cox_model_comp <- coxph(Surv(FupAfterFirstRecMonths, SecondRecOrDeath) ~ Age + Gender +
                          RecMultinodular + RecNoduleLargeSize + RecExtrahepatic +
                          TimeToFirstRecMonths + RecTreat, hcc)
```

```
## Warning in Surv(FupAfterFirstRecMonths, SecondRecOrDeath): Invalid status value,
## converted to NA
```

```
# Printing results using kable
result_comp <- finalfit::fit2df(cox_model_death, condense = FALSE)
kable(result_comp, booktabs = T,
      caption = "modello di Cox per endpoint composito") %>%
kable_styling(latex_options = c("striped", "HOLD_position"))
```

Table 5: modello di Cox per endpoint composito

explanatory	HR	L95	U95	p
Age	1.0157406	0.9738923	1.059387	0.4668782
GenderM	0.8854076	0.3972787	1.973291	0.7659684
RecMultinodular	1.2679008	0.5685631	2.827430	0.5618651
RecNoduleLargeSize	2.2379354	0.9102780	5.502006	0.0792345
RecExtrahepatic	1.5187318	0.6292372	3.665623	0.3526196
TimeToFirstRecMonths	0.9991399	0.9804004	1.018237	0.9290196
RecTreatPAL	1.2313122	0.5241823	2.892371	0.6329670

4.2 Visualizzazione grafica dei risultati con dei *forest plot*

Un modo efficiente per visualizzare graficamente l'effetto stimato di ciascuna variabile indipendente sul rischio di secondo evento competitivo (seconda recidiva o morte) o sull'endpoint composito è il *forest plot*. Questo grafico, infatti, può essere utilizzato per valutare l'importanza relativa delle diverse variabili indipendenti e la loro associazione con l'outcome. Il seguente codice implementa la creazione dei *forest plot* sfruttando la libreria *meta*, che permette di creare grafici di questo tipo a partire dai risultati dei modelli di regressione.
I *forest plot* associano ad ogni covariata un segmento la cui dimensione è dettata dall'intervallo di confidenza. In genere, se il segmento orizzontale attraversa la linea verticale (che rappresenta il valore 0), allora non c'è evidenza di un effetto significativo della variabile predittiva sull'outcome. Se il segmento orizzontale si trova sopra la linea, allora la variabile predittiva è associata a un aumento del rischio di evento, mentre se il segmento orizzontale si trova sotto la linea, la variabile predittiva è associata a una riduzione del rischio di evento.

4.2.1 *forest plot* modello di Cox seconda recidiva

Di seguito è riportato il codice *R* per la creazione del *forest plot* nel caso del modello di Cox per la seconda recidiva.

```
# Creating an object containing cox model results
cox_model_results <- summary(cox_model_rec)
# Extracting coefficients, confidence interval and p-value
coef <- cox_model_results$coef[,1]
lower <- cox_model_results$coef[,3]
upper <- cox_model_results$coef[,4]
pval <- cox_model_results$coef[,5]
# Creating dataframe containig the results
results_df <- data.frame(coef=coef, lower=lower, upper=upper, pval=pval,
                         row.names=names(coef))
# Sorting the results based on the p-value
results_df <- results_df[order(results_df$pval),]
# Crating forest plot using a meta function
meta::forest(x = results_df$coef,
             ci.lb = results_df$lower,
             ci.ub = results_df$upper,
             slab = rownames(results_df),
             psize = results_df$pval)
```

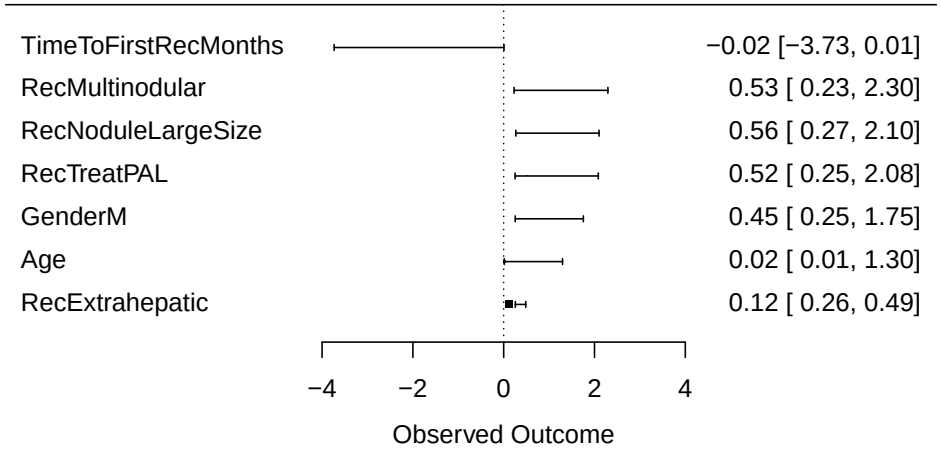

Figure 7: forest plot modello di Cox seconda recidiva

Dal primo *forest plot* si evince che tutte le variabili tranne *TimeToFirstRecMonths* influiscono attivamente sul rischio di seconda recidiva.

4.2.2 *forest plot* modello di Cox morte senza seconda recidiva

Nel caso del modello di Cox per la morte senza seconda recidiva i risultati che si ottengono sono i seguenti.

```
# Creating an object containing cox model results
cox_model_results <- summary(cox_model_death)
# Extracting coefficients, confidence interval and p-value
coef <- cox_model_results$coef[,1]
lower <- cox_model_results$coef[,3]
upper <- cox_model_results$coef[,4]
pval <- cox_model_results$coef[,5]
# Creating dataframe containig the results
results_df <- data.frame(coef=coef, lower=lower, upper=upper, pval=pval,
                         row.names=names(coef))
# Sorting the results based on the p-value
results_df <- results_df[order(results_df$pval),]
# Crating forest plot using a meta function
meta::forest(x = results_df$coef,
             ci.lb = results_df$lower,
             ci.ub = results_df$upper,
             slab = rownames(results_df),
             psize = results_df$pval)
```

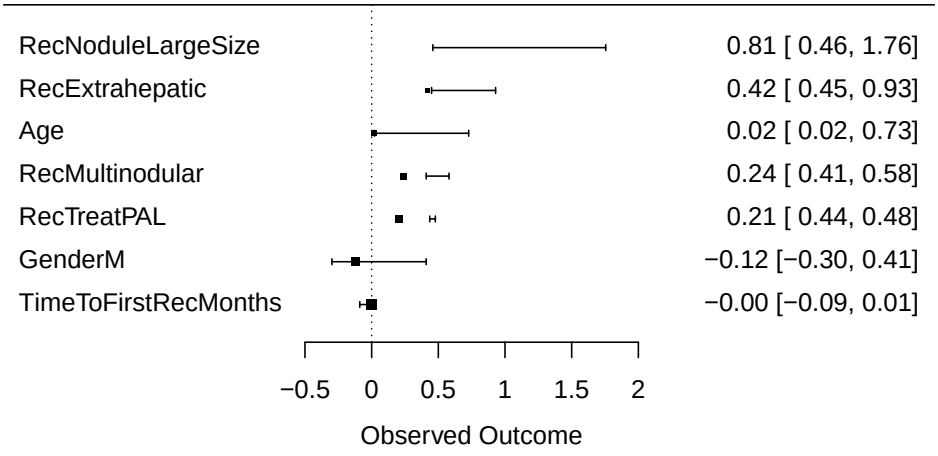

Figure 8: forest plot modello di Cox morte senza seconda recidiva

Da quest'altro *forest plot* si osserva che le covariate che influiscono maggiormente sono *RecNoduleLargeSize* e *RecExtrahepatic*.

4.2.3 *forest plot* modello di Cox endpoint composito

Infine questo è il caso del modello di Cox per l'endpoint composito.

```r
# Creating an object containing cox model results
cox_model_results <- summary(cox_model_comp)
# Extracting coefficients, confidence interval and p-value
coef <- cox_model_results$coef[,1]
lower <- cox_model_results$coef[,3]
upper <- cox_model_results$coef[,4]
pval <- cox_model_results$coef[,5]
# Creating dataframe containig the results
results_df <- data.frame(coef=coef, lower=lower, upper=upper, pval=pval,
                         row.names=names(coef))
# Sorting the results based on the p-value
results_df <- results_df[order(results_df$pval),]
# Crating forest plot using a meta function
meta::forest(x = results_df$coef,
             ci.lb = results_df$lower,
             ci.ub = results_df$upper,
             slab = rownames(results_df),
             psize = results_df$pval)
```

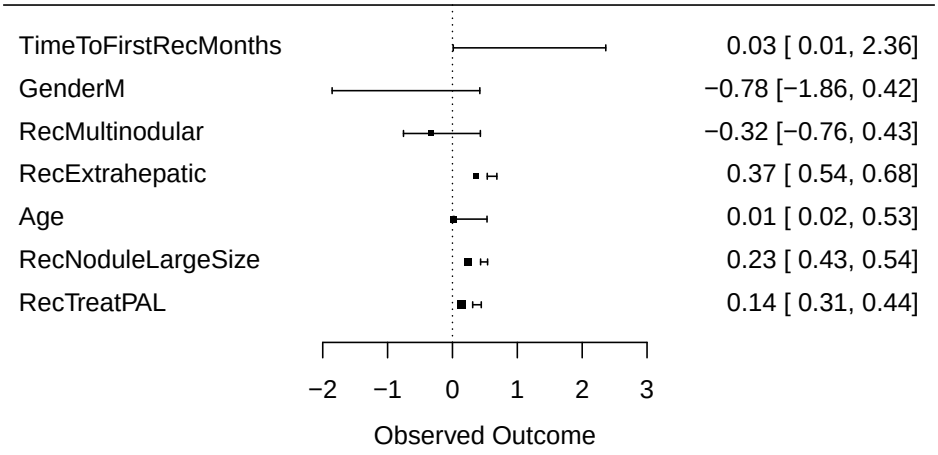

Figure 9: forest plot modello di Cox endpoint composito

Dal *forest plot* associato al modello di Cox per l'endpoint composito si osserva che le covariate *Gender* e *RecMultinodular* non influiscono sull'outcome dell'endpoint composito.

5 Modello predittivo

Il modello di Cox è definito come

$$h(t) = h_0(t) \cdot e^{(b_1 x_1 + b_2 x_2 + \ldots + b_n x_n)}$$

dove $h_0(t)$ è il *baseline hazard*. Il modello di Cox su R fornisce informazioni solo sul termine esponenziale $e^{(b_1x_1+b_2x_2+...+b_nx_n)}$ e, quindi, per costruire un modello predittivo e ottenere una funzione di sopravvivenza è necessario calcolare il *baseline hazard*. Per fare questo si possono sfruttare la funzione *basehaz()* o anche la funzione *survfit()* (è stata scelta la seconda opzione). Il codice R che implementa questo passaggio e calcola la funzione di sopravvivenza per l'endpoint composito è il seguente.

```
# Calling survfit funcion
fit<-survfit(Surv(FupAfterFirstRecMonths, SecondRecOrDeath) ~ 1, hcc)
# Plotting survival probability with ggsurvplot
ggsurvplot(fit, conf.int = TRUE,
           xlab = 'Months',
           ylab='Survival probability',
           legend = "none",
           ggtheme = theme_minimal())
```

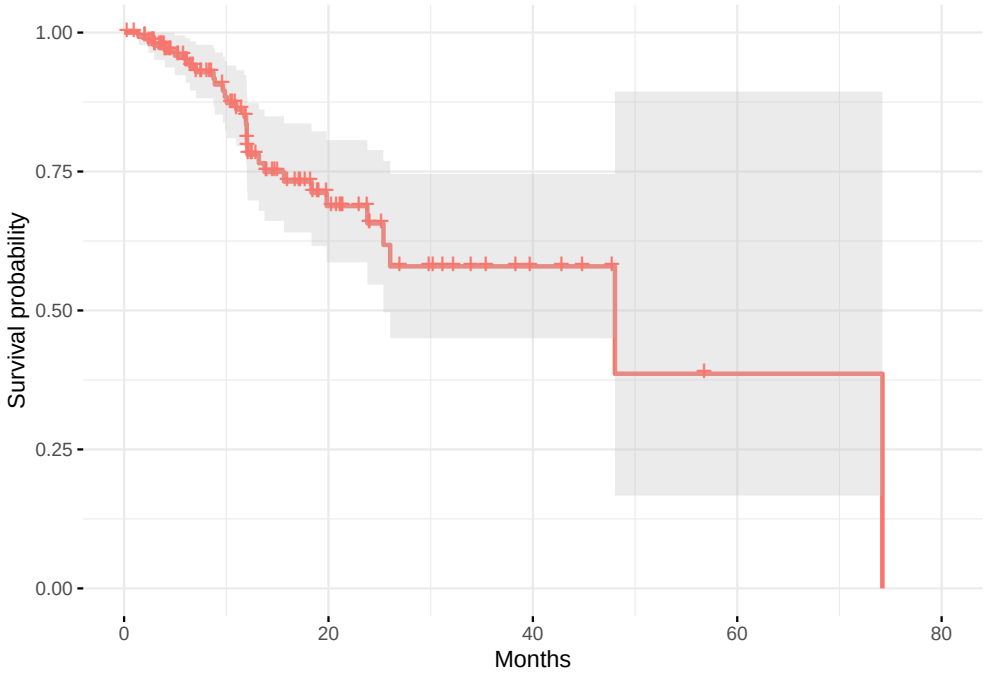

Figure 10: survival probability per endpoint composito

La funzione di sopravvivenza esprime, per l'appunto, la probabilità di survival in funzione del tempo. In questo caso il tempo è espresso come mesi a partire dalla prima recidiva.

6 Valutazione delle assunzioni

Il *modello di Cox* si basa su delle assunzioni come la proporzionalità degli azzardi e la linearità delle variabili. Per questa ragione è necessario verificare le assunzioni per dare affidabilità al modello.

6.1 Valutazione forma funzionale delle variabili continue

È possibile valutare la forma funzionale delle variabili continua (*Age* e *TimeToFirstRecMonths*). Ci si chiede, quindi, se esiste una relazione non lineare tra la variabile stessa e la variabile di risposta. Ciò può essere fatto attraverso l'analisi di grafici che rappresentano la relazione tra la variabile indipendente (la variabile

continua) e l'azzardo. Se la relazione tra le due variabili è lineare, ci si aspetta di osservare un andamento lineare dell'azzardo lungo tutto l'intervallo di valori della variabile indipendente. Se invece la relazione è non lineare, si possono osservare curve a campana o altre forme particolari.

```
# Setting canvas
par(mar=c(4,4,1,1), mfrow=c(1,2))
# Generating first model and plotting graph
model_plot_Age <- coxph(Surv(FupAfterFirstRecMonths, SecondRecOrDeath) ~ bs(Age),
                        data = hcc)
plotHR(model_plot_Age, term="Age",
       plot.bty="o", ylog=T,
       xlim = c(0, 100),
       rug="density",
       main = "Age",
       polygon_ci=T)
# Generating second model and plotting graph
model_plot_TimeToFirstRecMonths <- coxph(Surv(FupAfterFirstRecMonths,
                                              SecondRecOrDeath) ~ bs(TimeToFirstRecMonths),
                                         data = hcc)
plotHR(model_plot_TimeToFirstRecMonths, term="TimeToFirstRecMonths",
       plot.bty="o", ylog=T, xlim = c(0, 300),
       rug="density", polygon_ci=T,
       main = "TimeToFirstRecMonths")
```

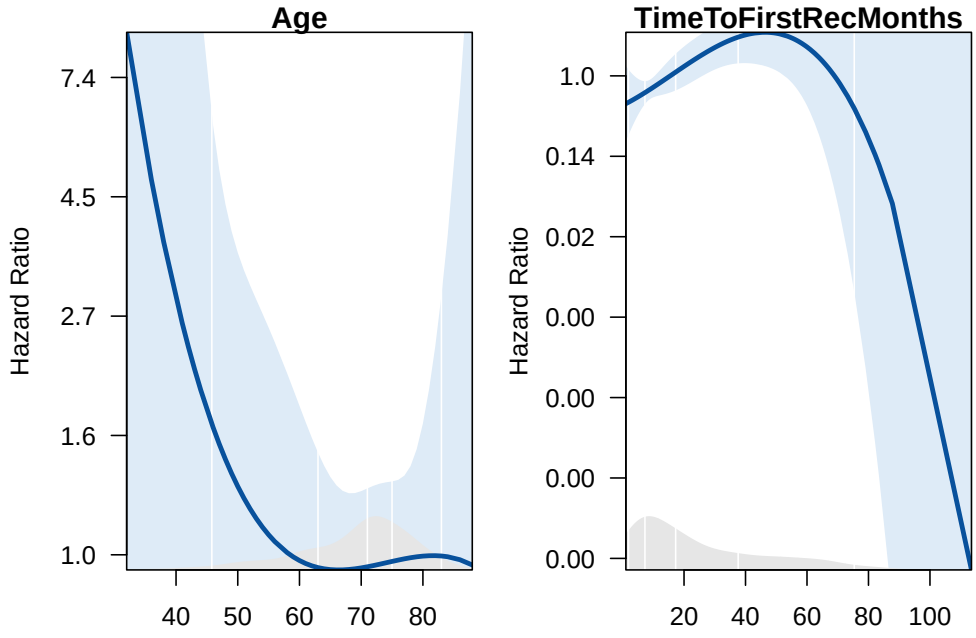

Figure 11: linearity assumption

Di seguito si mostrano anche i *residui di Martingale*. Questi grafici mostrano i residui in funzione della covariata continua che si sta studiando. Anche in questo caso ci si aspetta di trovare un andamento lineare se l'assunzione fosse verificata ma, come si può osservare, non sarà così. Per il grafico dei *residui di Martingale* è stato sfruttato il comando *ggcoxfunctional()*.

```
# Martingale residuals (Age e TimeTOFirstRecMonths)
ggcoxfunctional(Surv(FupAfterFirstRecMonths, SecondRecOrDeath) ~
                  Age + TimeToFirstRecMonths, data = hcc)
```

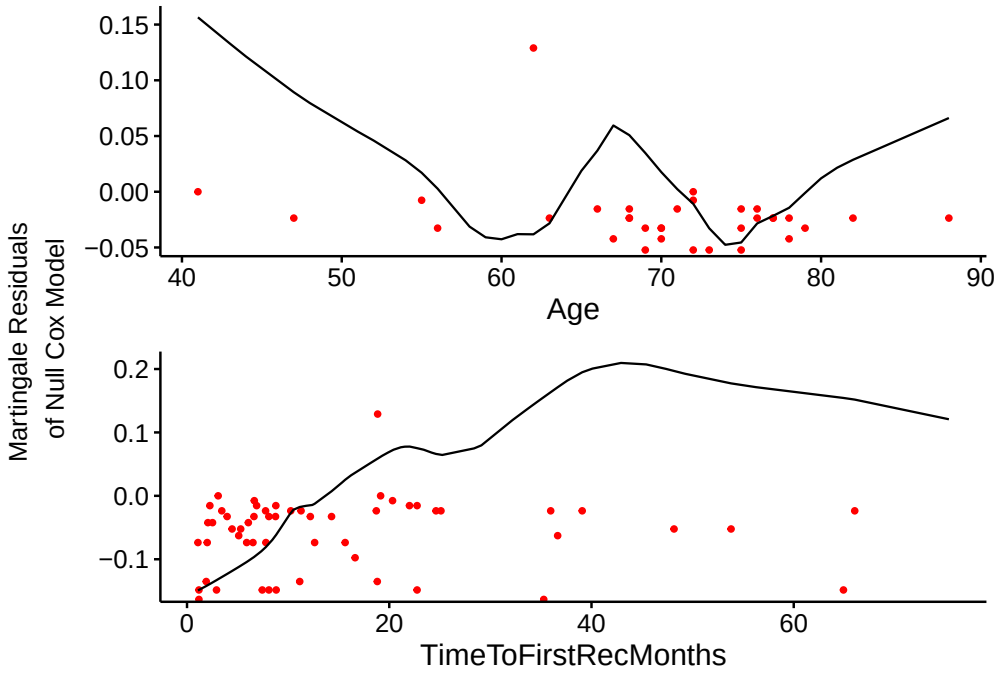

Figure 12: Martingale residuals

Guardando i grafici, si ossserva che la condizione di linearità non sembra essere soddisfatta pertanto, come si verificherà in seguito, le performance del modello non saranno molto elevate.

6.2 Assunzione *Proportional Hazards*

Per valutare l'assunzione di proporzionalità degli hazard, si può utilizzare il *test di Schoenfeld*, che verifica se l'effetto delle covariate sul rischio di evento è costante nel tempo. Il test confronta i *residui di Schoenfeld*, ovvero la differenza tra il valore osservato della covariata e il valore atteso sotto l'ipotesi di proporzionalità degli hazard, in funzione del tempo. Se non vi è alcuna tendenza sistematica, ovvero i residui non dipendono dal tempo, allora l'assunzione di proporzionalità degli hazard può essere considerata valida. Il seguente codice implementa il *test di Schoenfeld*.

```
# Test di proporzionalità degli hazard
checkPH <- cox.zph(cox_model_comp)
# Converting result into data frame
checkPH_df <- as.data.frame(checkPH$table)
#Printing results with kable
kable(checkPH_df, booktabs = T, caption = "test di Schoenfeld") %>%
kable_styling(latex_options = c("striped", "HOLD_position"))
```

Table 6: test di Schoenfeld

	chisq	df	p
Age	0.1162247	1	0.7331658
Gender	3.9502206	1	0.0468652
RecMultinodular	0.3339065	1	0.5633678
RecNoduleLargeSize	0.1237022	1	0.7250534
RecExtrahepatic	0.8410857	1	0.3590864
TimeToFirstRecMonths	0.0932118	1	0.7601330
RecTreat	0.1234146	1	0.7253602
GLOBAL	5.3027781	7	0.6230665

Assumendo come riferimento un *p value* di 0.05, si puó stabilire che se $pvalue < 0.05$ allora l'ipotesi di proporzionalitá deve essere rigettata mentre se $pvalue > 0.05$ l'ipotesi puó essere accettata. In questo caso, tutti i *pvalue* rispettano l'ipotesi di proporzionalitá.
Si possono visualizzare graficamente i residui, come mostrato nel seguente codice *R*.

```
# Stampo i risultati con la funzione ggcoxph della libreria survminer - 1
ggcoxzph(checkPH[1:4])
```

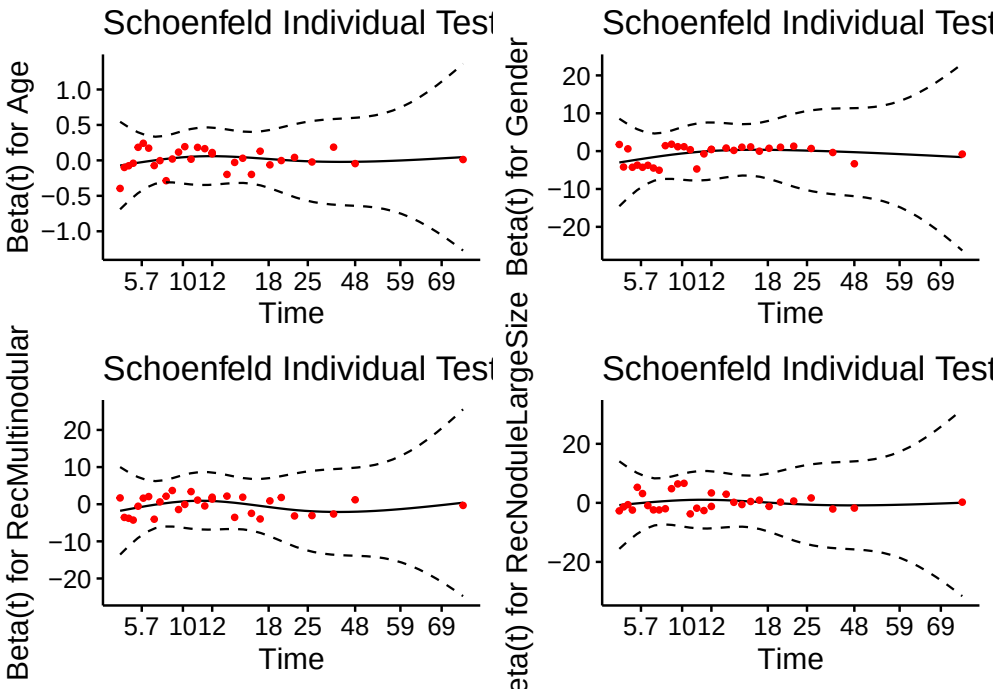

Figure 13: valutazione assunzione proportional hazards - 1

```
# Stampo i risultati con la funzione ggcoxph della libreria survminer - 2
ggcoxzph(checkPH[5:7])
```

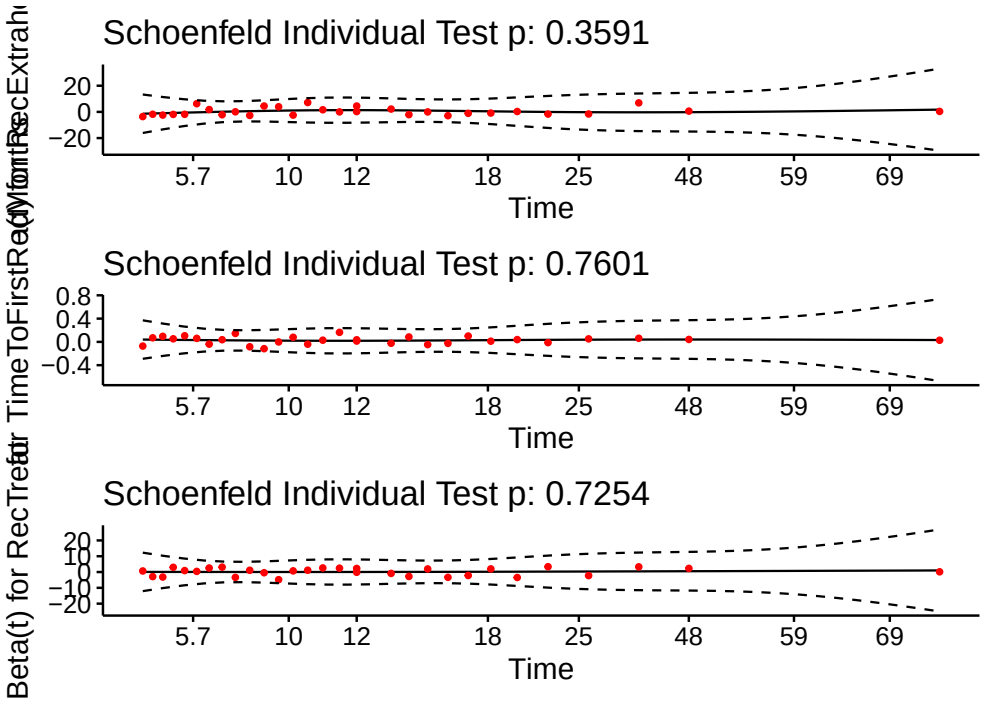

Figure 14: valutazione assunzione proportional hazards - 2

7 Valutazione delle performance

A questo punto è necessario valutare le performance del modello. La valutazione può essere svolta sfruttando la creazione di tre differenti grafici che valutano: *discriminazione*, *calibrazione* e *net benefit*.

```
# Computing model for performance evaluation
model_eval <- coxph(formula = Surv(FupAfterFirstRecMonths,
                                   SecondRecOrDeath) ~ Age + Gender + RecMultinodular +
                      RecNoduleLargeSize + RecExtrahepatic +
                      TimeToFirstRecMonths + RecTreat,
                    data = hcc, x=TRUE)
# Calling survfit function
fit <- survfit(model_eval, newdata = hcc)
hcc$risk<-1-as.numeric(summary(fit,times=36)$surv)
# Calling score function from riskRegression library
score<- Score(list("model1" = model_eval),
              formula = Surv(FupAfterFirstRecMonths, SecondRecOrDeath)~1,
              data = hcc, conf.int = T,
              times = 36,
              plots = c("calibration","ROC"))
```

7.1 Calibration plot

In *calibration plot* confronta la proporzione di eventi osservati con la proporzione di eventi predetti dal modello a diversi livelli di probabilità stimata. Se il modello è ben calibrato, il *calibration plot* mostrerà una linea diagonale perfetta, che indica che i valori osservati e stimati sono perfettamente allineati. Il seguente codice *R* implementa la creazione del calibration plot.

```
# Plotting calibration plot
plotCalibration(score,cens.method="local",method="quantile",q=10)
title(main="calibration at 3 years")
```

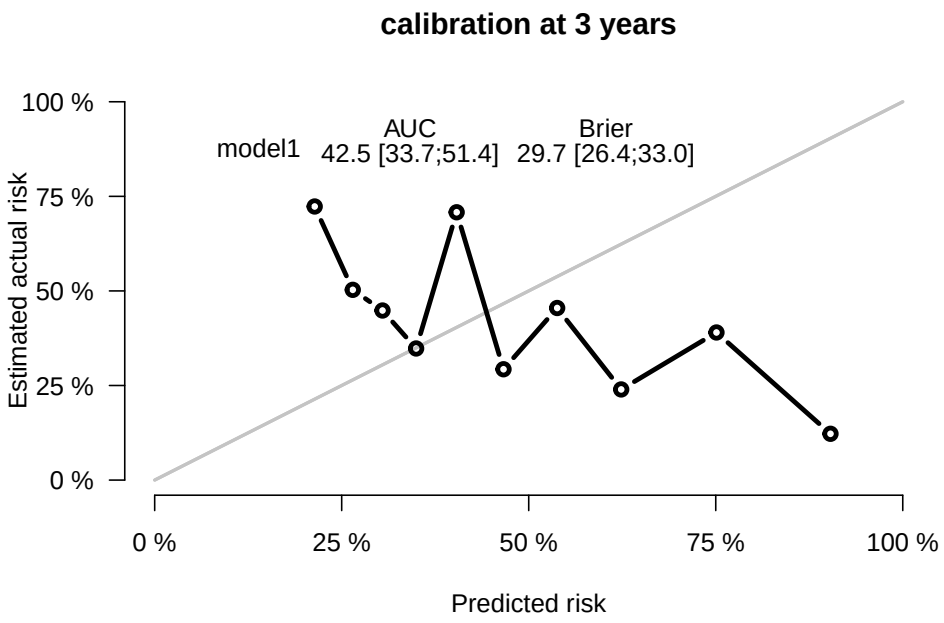

Figure 15: calibration plot

Come ci si aspettava dopo aver visto che l'assunzione di linearità non è verificata, le performance del modello non sembrano essere buone.

7.2 ROC curve

```
# Plotting ROC curve
plotROC(score,cens.method="local")
title(main = "time-dependent ROC at 36 months")
```

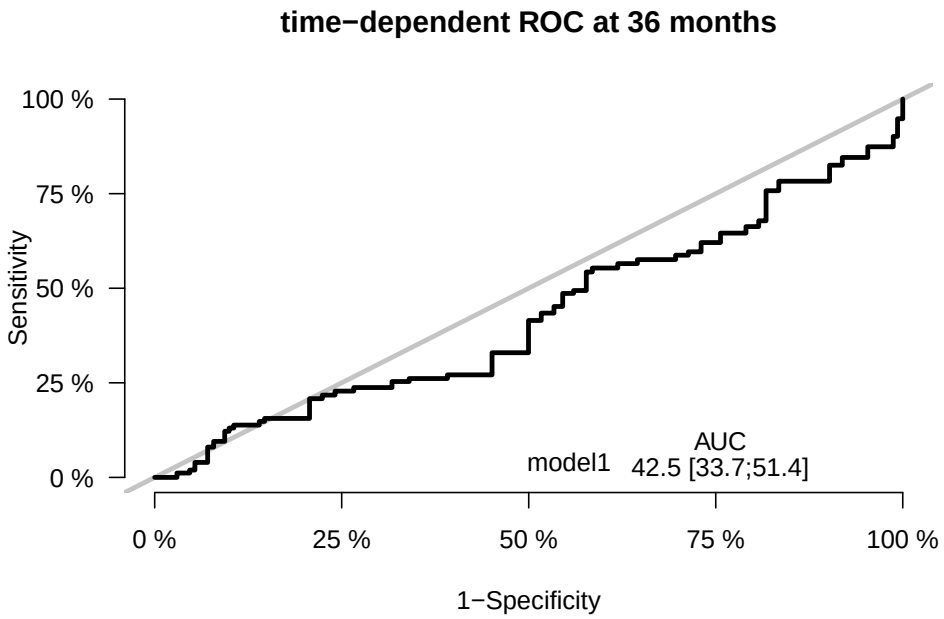

Figure 16: ROC curve

Anche la ROC curve conferma quanto detto in precedenza dal momento che l'AUC è addirittura inferiore al 50% (infatti è pari al 42.5%). Il modello è tanto più efficiente nella discriminazione quanto più l'AUC si avvicina al 100%.

7.3 Net Benefit

```
# Computing Net Benefit
dca(Surv(FupAfterFirstRecMonths, SecondRecOrDeath) ~ risk,
    data = hcc, time = 36)
```

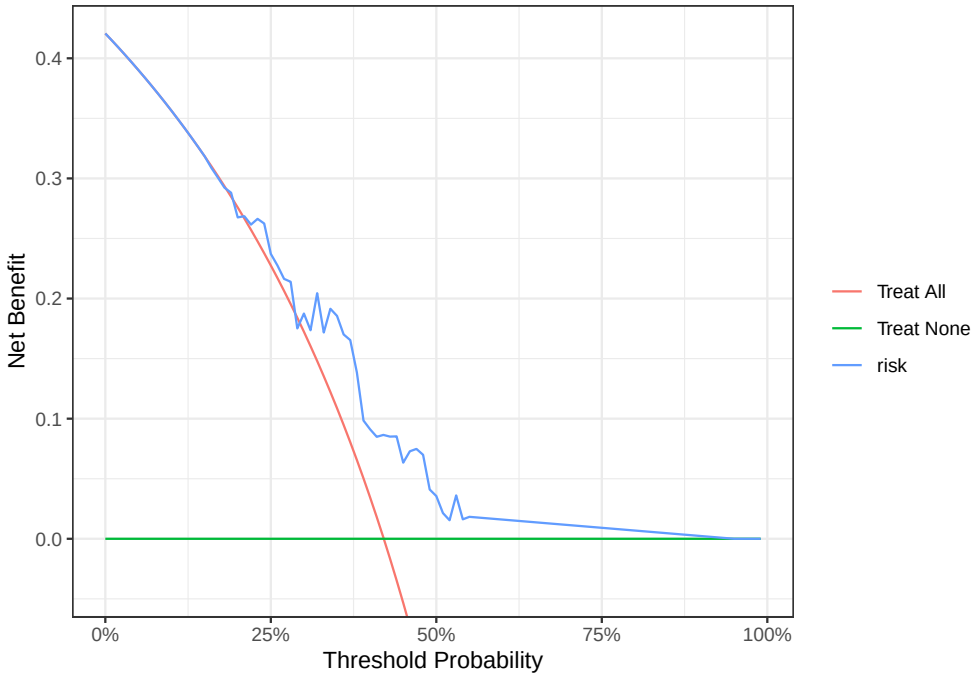

Figure 17: Net Benefit

Il grafico del *Net Benefit* indica che il modello predittivo è efficiente fintantochè la curva rimane sopra le due che fanno riferimento a *Threat None* e *Threat All* (ovvero due strategie opposte). Per soglie di probabilità molto alte ovviamente la curva del net benefit tende ad appiattirsi e si avvicina a quella di *Threat None* mentre per soglie basse la curva ha un andamento molto simile a quello di *Threat All*. Si possono notare delle piccole regioni in cui il *Net benefit* è al di sotto della soglia del *Threat All*.

8 Predizione del rischio

Sono stati ipotizzati i dati di due pazienti (un paziente giovane e uno anziano) ed è stato estratto casualmente un paziente dal dataset sfruttando il comando *sample()*. Per ciascuno di questi tre pazienti è stata calcolata la probabilità di evento a 36 mesi dalla comparsa della prima recidiva. Il seguente codice implementa il calcolo della probabilità e stampa una tabella con i risultati.

```
# Peggior combinazione per un giovane
first <-survfit(cox_model_comp, newdata=data.frame(Age = 16,
                                                   Gender = "M",
                                                   RecMultinodular = 1,
                                                   RecNoduleLargeSize = 1,
                                                   RecExtrahepatic = 1,
                                                   RecTreat = "PAL",
                                                   TimeToFirstRecMonths = 32.4))
# Miglior combinazione per un anziano
second <-survfit(cox_model_comp, newdata=data.frame(Age = 86,
                                                    Gender = "F",
                                                    RecMultinodular = 0,
                                                    RecNoduleLargeSize = 0,
                                                    RecExtrahepatic = 0,
                                                    RecTreat = "CUR",
```

```
                                                    TimeToFirstRecMonths = 32.4))
# Estrazione casuale nel dataset
numero = sample(1:317, 1)
third <- survfit(cox_model_comp, newdata = hcc[numero, ])
# Creazione tabella con risultati
tabella_prob <- data.frame(
  Paziente = c("giovane", "anziano", "casuale"),
  Probabilità = c(1- summary(first,times=36)$surv,
                  1- summary(second,times=36)$surv,
                  1- summary(third,times=36)$surv)
)
# Printing results with kable
kable(tabella_prob, booktabs = T, caption = "probabilità di evento per tre pazienti") %>%
kable_styling(latex_options = c("striped", "HOLD_position"))
```

Table 7: probabilità di evento per tre pazienti

Paziente	Probabilità
giovane	0.4056154
anziano	0.8246094
casuale	0.6065347

9 Conclusioni

Il modello predittivo che è stato creato sfruttando il *modello di Cox* non sembra essere efficiente e questo è dovuto principalmente al fatto che l'assunzione di linearità delle variabili continue non è soddisfatta. É stato comunque possibile effettuare l'analisi e ottenere delle probabilità di evento per dei pazienti ipotetici.
Il dataset fornito è di numerosità non elevata (solamente 317 pazienti) e con un numero modesto di covariate: sarebbbe interessante compiere un indagine su un quadro clinico dei soggetti di studio più ampio, per esempio relativo alla presenza o assenza di eventuali malattie pregresse o all'esposizione di alcuin fattori di rischio comuni nei soggetti affetti da carcinoma epatocellulare, come la cirrosi epatica, la presenza di virus epatite B o C oppure la presenza di α-fetoproteina.
Il dataset, dunque potrebbe essere non del tutto completo per i fini di uno studio clinico ma può essere comunque utile per fornire informazioni preliminari ad ulteriori studi, infatti le capacità di previsione del modello creato è generalmente poco soddisfacente ma fornisce spunti interessanti per ulteriori analisi.