Introduction to Survival Analysis

Solution to Exercises

The R scripts with the solutions are provided together with the resource materials. These are the files called :

***solutions\_survival\_analysis\_ex\_n1.R***

***solutions\_survival\_analysis\_ex\_n2.R***

Exercise Set #1

Use the dataset provided to complete the following tasks:

1. Create a survival object through the R package “survival” and considering the discrete timing of the target event in the provided dataset (i.e.: variable “timedis”). Check that the survival object has been correctly specified.

After loading the dataset and called it “df”, I have inspected the variables using the standard command:

head(df)

This should highlight that in the dataset there is no variable for the event occurrence. This can be easily created considering that, just like other similar survival analysis datasets, the target event was not recorded for participants that are censored (either because they left the study before the end of the study, or because the study stopped at some point). Therefore, we can create an “event” indicator by using the information in the “censor” variable:

df$event <- ifelse(df$censor == 1, 0, 1)

The line above creates a variable “event” within the dataset “df” considering that if the variable “censor” in the “df” dataset is equal to 1, “event” will be equal to 0. The ***ifelse*** function also considers that if the statement is false (i.e. censor = 1 is false, hence censor is not equal to 1), then “event” will be equal to 1.

Having created the “event” variable, I created a survival object using the ***Surv*** command from the ***survival*** library:

surv\_obj <- Surv(time = df$timedis, event = df$event)

The line above instructs R to consider the timing of the event the variable “timedis” from dataset “df” and the event indicator the variable “event” from dataset “df”.

To check R is correctly reading the survival object, you can inspect a few cases in it:

surv\_obj[1:15]

which produces this output:



These represent the first 15 cases in the dataset and their timing of the event in the discrete time units (variable “timedis”), unless the timing is followed by a “+” sign, which indicates the case is censored.

1. Create a life table for the whole sample based on the discrete time variable. Ensure that the life table reports the risk set, the number of cases who experienced the event in each time interval, the number of cases censored in each time interval, the survival and the hazard function.

Using the ***survfit*** function from the ***survival*** package, it is possible to extract most of the information needed for the life table. The argument of the function is the survival object that has just been created in exercise 1:

lti <- survfit(surv\_obj ~ 1, data = df)

The “lti” object now includes most of the information needed for the life table. The only information that is not reported is the hazard function, but this can be easily calculated from the information in “lti”:

hazard\_function <- lti$n.event / lti$n.risk

All the information can now be collated into a data frame I called “lt” (for life table):

lt <- data.frame(

time = lti$time,

n.risk = lti$n.risk,

n.event = lti$n.event,

n.censor = lti$n.censor,

survival = round(lti$surv, 3),

hazard = round(hazard\_function, 3)

)

Note that in the last two lines I have rounded the survival and hazard function to only display three decimals.

To edit this table, I have used the “kable” function:

lt %>%

kbl() %>%

kable\_classic\_2(full\_width = F)

obtaining this output:

A screenshot of a computer

Description automatically generated

1. Plot the hazard function, preferably using the “ggplot” package.

The information in the life-table object “lt” created above can be used to plot the hazard function. To this aim, I have used ***ggplot*** package, and the ***geom\_line*** option to draw a line that connects the hazard function estimates:

*# Plot the discrete time hazard function with color palette*

plot\_dth<-ggplot(lt, aes(x = time, y = hazard)) +

geom\_line(linewidth=3, col="brown") +

labs(title = "Discrete Time Hazard Function",

x = "Time",

y = "Hazard Function") +

ylim(c(0,.5)) +

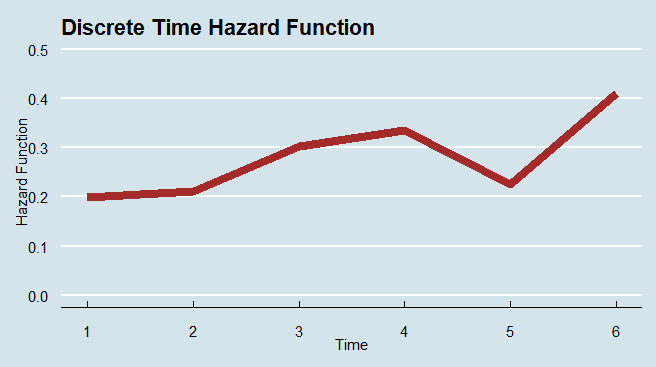
theme\_economist() +

scale\_x\_continuous(breaks =

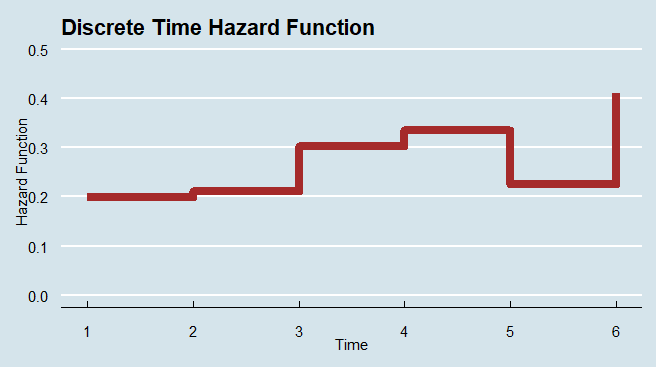
seq(1, 6, by = 1)) #changes the ticks in x axis

plot\_dth

The first two lines in the script above are the key ones because the instruct R to create a plot based on the data in data frame “lt”, where the *x* axis is going to represent the time values, the *y* axis the hazard function values, and ***geom\_line*** instructs R to plot a line to connect the hazard values. The other lines are instructions concerning labels in the plot and its aesthetic. For example, I used the “Economist” theme:

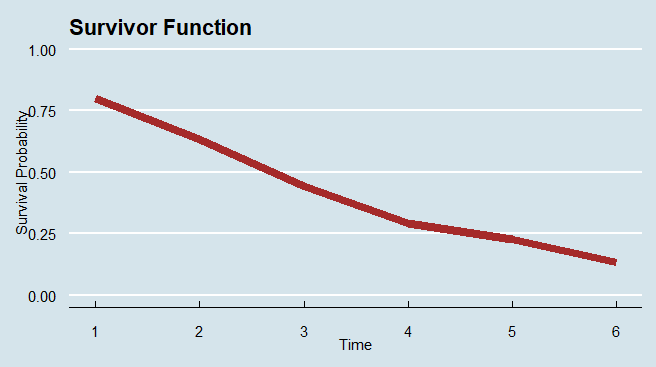
****

**NOTE:** I have used lines to connect the estimate of the hazard and survival functions, but other authors prefer to represent these estimates as steps when the time units are discrete, as in this case (see the example below, which I created using ***geom\_step*** option instead of ***geom\_line*** in ***ggplot***.) I prefer to use lines because these generally make it easier to detect a pattern.



1. Plot the survival function, preferably using the “ggplot” package.

The steps for creating this plot are the same as for the hazard function plot: using the data frame “lt”, I selected the variables of interest and used a line to connect the estimates of the survivor function. The other lines provide some aesthetic choices:



1. Calculate the median lifetime of survival function using interpolation, and add this in the survival function plot.

If we look at the life-table reported above, we can see that the median lifetime of the sample follows between time interval 2 (survivor function: 0.632) and time interval 3 (survivor function: 0.441). These time intervals represent periods of two months, so the median lifetime falls between 4 and 6 months.

We can use interpolation to estimate a more precise time point where we would expect half of the same had experienced the target event (relapse). The interpolation is between the two values of the estimated survival function for a time interval *j* where the value of the survival function is above .5, and the following time interval j+1, where the estimated survival function is less than .50.

The formula for the linear interpolation is as follows:

Where *m* represents the time interval where the survival function is >.5, and *m+1* represents the following time interval where the survival function is <.5. represents the estimated survival function in interval *m*, and represents the estimated survival function in interval *m+1*.

If we consider the data from the life table reported above and substitute with the estimated values we obtain:

2.69

In the exercises, I had calculated this automatically in this way:

*# Extract survival probabilities and times*

surv\_probs <- summary(lti)$surv

surv\_times <- summary(lti)$time

*# Find the time points around the median survival probability (0.5)*

lower\_index <- max(which(surv\_probs >= 0.5))

upper\_index <- min(which(surv\_probs < 0.5))

*# Linear interpolation to find the median lifetime*

median\_lifetime <- surv\_times[lower\_index] +

((surv\_probs[lower\_index]-.5) /

(surv\_probs[lower\_index] - surv\_probs[upper\_index]))

*# Print the median lifetime*

print(median\_lifetime)

I can now add a dashed line representing the median lifetime value to the plot of the survivor function, as well as some annotation:

plot\_surf2<-plot\_surf+

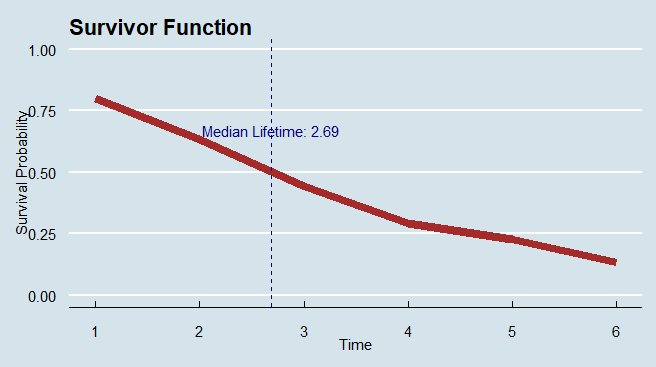
geom\_vline(xintercept = median\_lifetime, linetype = "dashed", color = "navy") +

annotate("text", x = median\_lifetime, y = 0.5,

label = paste("Median Lifetime: 2.69"),

vjust = -3, color = "navy")

plot\_surf2



Since the time intervals correspond to 2 months of time, (2.69\*2) corresponds to approximately 5.38 months: we expect that passed the fifth month after the intervention, 50% of the adolescents will have relapsed and abused alcohol.

1. Compare the two plots to draw some insights.

I have used the ***ggpubr*** package to combine the two plots into a single figure:

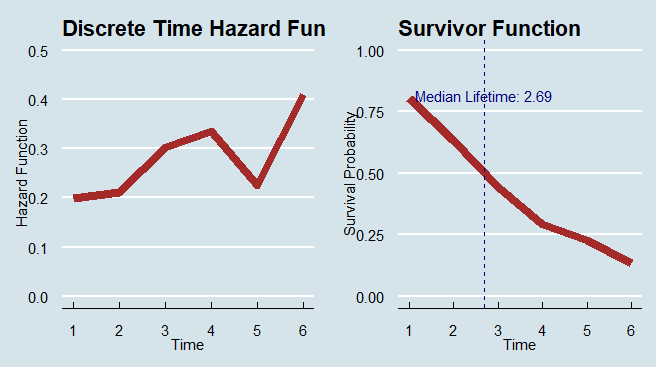
*### Combine the plots in figure ####*

library(ggpubr) #to combine plots

figure <- ggarrange(plot\_dth, plot\_surf2,

ncol = 2)

figure



The two plots suggest that risk for relapse is relatively stable in the first 4 bi-monthly periods: this stability is reflected in the survivor function, which seems to follow a linear trajectory in the first four time intervals. There appears to be a trough in risk for relapse during the 5th time interval, which is reflected in the survivor function line being shallower. Finally, there is an increase in risk for relapse during the final time interval.

Exercise Set #2

Use the dataset provided to complete the following tasks:

1. Create life tables for females and males in the dataset considering the discrete time variable (“timedis”). Ensure the life tables report the risk set, the number of cases who experienced the event in each time interval, the number of cases censored in each time interval, the survival and the hazard function.

Similarly the previous exercise set, I will use the ***survival*** package and its ***Surv*** function to create survival objects, but this time the ***Surv*** function is used as an argument within the ***survfit***function:

*#estimate the information for each subset based on Male covariate*

lts1 <- survfit( Surv(timedis, event)~ 1, conf.type="none",

subset=(male==0), data=df)

lts2 <- survfit( Surv(timedis, event)~ 1, conf.type="none",

subset=(male==1), data=df)

The script above creates two object “lts1” and “lts2” which represent the estimated survivor functions and other survival analysis information for females and males, respectively. To ensure that “lts1” only includes females, I added the ***subset=(male==0)*** option to the ***survfit*** command. Using a similar option I ensured that “lts2” only includes males.

*Note*: An alternative way to stratify the estimates would have involved specifying the covariate after the ***~*** in the ***survfit*** function:

lti <- survfit( Surv(timedis, event)~ male, conf.type="none",

data=df)

By substituting ***~1*** with ***~male***, I am asking to display estimated stratified by values of the covariate male, rather than for the whole sample. I chose to create two data frames, one for each level of covariate male, because of convenience (I find it makes it slightly easier to handle the data for plotting).

Similarly to the example the previous set of examples, I then calculated the hazard functions in each time period for females and males using the information in data frames “lts1” and “lts2”, respectively:

*# calculate the hazard functions*

h1<-lts1$n.event/lts1$n.risk

h2<-lts2$n.event/lts2$n.risk

The hazard functions are not estimated by ***survfit***, but I can easily estimate them based on the information on the number of events and the risk set in each time period.

Finally, I can collate all the information into two data frames:

tab3\_1<-data.frame(time=lts1$time, n.risk=lts1$n.risk,

n.event=lts1$n.event, n.censor=lts1$n.censor,

hazard=round(h1, 3), survival=round(lts1$surv, 3))

tab3\_2<-data.frame(time=lts2$time, n.risk=lts2$n.risk,

n.event=lts2$n.event, n.censor=lts2$n.censor,

hazard=round(h2,3), survival=round(lts2$surv, 3))

and display the tables using the ***kable*** function:

A table with numbers and symbols

Description automatically generated A table with numbers and symbols

Description automatically generated

1. Plot the hazard functions for females and males in the same plot, preferably using the “ggplot” package.

In the previous exercise I have created two data frames, “tab3\_1” and “tab3\_2” that include survival analysis estimates for females and males respectively. In order to plot these functions, I will first extract the information I need from the two data frames:

*#select the values I need for the plots*

dth31<-select(tab3\_1, c("time", "hazard"))

dth32<-select(tab3\_2, c("time", "hazard"))

I then create a single data frame where I append the “time” and “hazard” values for females first, and then males:

*# Append values from the selection before*

dthall<-rbind(dth31, dth32)

I then add an indicator for sex to the latter dataset (there are only 6 discrete time periods for which hazard and survival functions are estimated, so I just need to ensure the first 6 rows are identified as estimates concerning females, and the remaining are recognised as estimates concerning males):

*#create indicator for male covariate*

dthall$male <- c(rep(0, 6), rep(1, 6))

Finally, I can use ggplot to plot the hazard function estimates and use lines to link these values:

*# Plot the discrete time hazard function with colour palette*

plot\_hmale<-ggplot(dthall, aes(x = time,

y = hazard, color = factor(male))) +

geom\_line(linewidth=3 ) +

scale\_color\_brewer(palette="Set2",labels = c("Female", "Male")) +

labs(title = "Hazard Function",

x = "Time",

y = "Hazard Function",

color = "Male") +

ylim(c(0,.5)) +

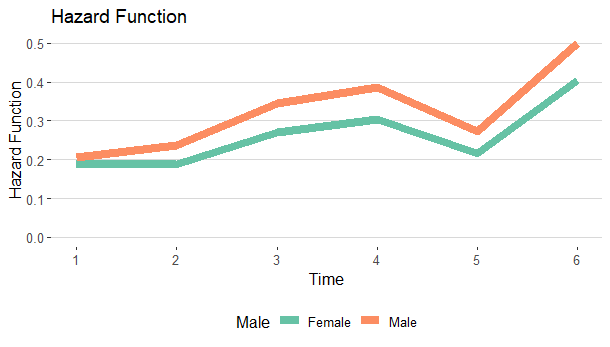
theme\_hc() +

scale\_x\_continuous(breaks =

seq(1, 6, by = 1)) #changes the ticks in x axis

plot\_hmale

The main different from Set #1 of the Exercises is that I added a ***color=factor(male)*** option to the ***aes*** function in ***ggplot***, which ensures that values relative to the levels of the covariate *male* are being plotted using different colours. As in previous example, using ***geom\_line*** ensures the values are linked using lines. The other options concern aesthetical choices (for example, I used the colour palette “Set 2” for the lines, which a preset option in the ***RColorBrewer*** library).



1. Plot the survival function for females and male in the same plot, preferably using the “ggplot” package. Add the median lifetime for females and males using interpolation

Firstly, I used linear interpolation to identify the exact median lifetime for the two groups. Inspection of the life tables indicate this will be between time period 2 and time period 3 for both females and males: the probability of surviving for a random participant for females and males is 50% at some point between time period 2 and time period 3.

In the first set of exercises I indicated the formula to calculate the exact median lifetime using linear interpolation. I also provided some code to calculate this automatically. For example, the median lifetime for females will be calculated by:

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Females

*# Extract survival probabilities and periods*

surv\_probs1 <- summary(lts1)$surv

surv\_times1 <- summary(lts1)$time

*# Find the time points around the median survival probability (0.5)*

lower\_index1 <- max(which(surv\_probs1 >= 0.5))

upper\_index1 <- min(which(surv\_probs1 < 0.5))

*# Linear interpolation to find the median lifetime*

median\_lifetime1 <- surv\_times1[lower\_index1] +

((surv\_probs1[lower\_index1]-.5) /

(surv\_probs1[lower\_index1] - surv\_probs1[upper\_index1]))

*# Print the median lifetime*

print(median\_lifetime1)

I can now collate the information concerning time periods, survivor functions, and covariates into a single dataset, similarly to what I did with the hazard function plots.

I can then draw the plots of the survivor function for females and males.

In order to add lines the represented the respective median lifetimes, I used this script:

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

median\_lifetimes <- data.frame(

male = c("Female", "Male"),

median\_lifetime = c(2.88, 2.50)

)

####\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

####\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*### Survival Curve with median lifetimes ####*

####\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

*# Add annotations for each Male group*

for (i in 1:nrow(median\_lifetimes)) {

plot\_surmale <- plot\_surmale +

geom\_vline(xintercept = median\_lifetimes$median\_lifetime[i], linetype = "dashed", color = scales::brewer\_pal(palette = "Set2")(3)[i]) +

annotate("text", x = median\_lifetimes$median\_lifetime[i],

y = 0.5,

label = paste("", median\_lifetimes$median\_lifetime[i]),

vjust = -5, color = scales::brewer\_pal(palette = "Set2")(3)[i])

}

*# Print the plot*

print(plot\_surmale)

which is adding dashed lines according to the *i* values in rows of the dataframe “median\_lifetimes” I have created to report the estimated values for females and males respectively.

1. Transform the hazard functions for females and males into odds and into logits, plot these transformed outcomes, and inspect the plots.

I used the data frame created in exercise #2 of this set of exercises: this data frame included the hazard functions for females and males. The odds are calculated according to the formula:

The logits are the natural logarithm of these odds. Therefore the relevant transformations are as follow:

*# odds transformation*

dthall$oddsh<-dthall$hazard/(1-dthall$hazard)

*# logit transformation*

dthall$logith<-log(dthall$oddsh) *#the natural log of the odds*

Once I have calculated these values, I can plot them using similar commands I used while plotting the hazard functions for females and males, but see the R scripts of the solutions for more details.

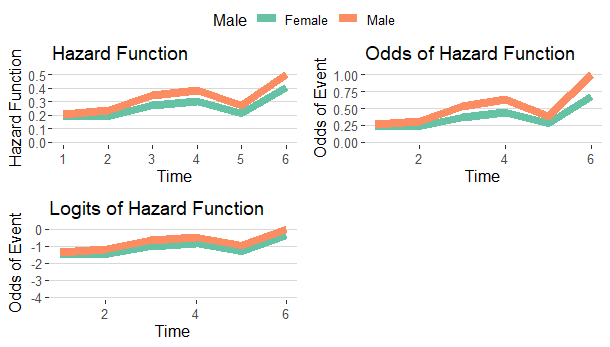
Finally, I can combine the plots in a single figure:

library(ggpubr) *#allows to arrange plots together*

figure <- ggarrange(plot\_hmale, plot\_oddmale, plot\_logitmale,

ncol = 2, nrow = 2, common.legend = TRUE, legend="top")

figure



The figure suggest it may be reasonable to assume that the logits of the hazard functions for females and males have the same shape across time and their distance may be constant in each time period.

1. Use a the person-period dataset “sa\_exercise\_pp.csv” to fit a general specification model of the logit hazard functions for females and males over discrete time periods. Use a logistic regression to fit the model. Describe and interpret the results: in particular, how can you interpret the parameter of the covariate “male”?

The “sa\_exercise\_pp.csv” dataset is the transposition of the dataset “sa\_exercise.csv” into a person-period format. In the scripts with the solution to the exercises, I have also written a script to carry out this transposition:

*####\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*####\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

*### Restructure data to Person-Period ####*

*####\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\**

library(tidyverse) # to manipulate dataset

*# Function to create person-period data*

create\_person\_period <- function(data) {

data %>%

rowwise() %>%

mutate(periods = list(1:timedis)) %>%

unnest(periods) %>%

mutate(event = ifelse(periods == timedis & event == 1, 1, 0),

censor = ifelse(periods == timedis & censor == 1, 1, 0)) %>%

filter(periods <= timedis) %>%

select(ID, periods, event, censor, male, ses)

}

The script above does the following operations:

1. mutate(periods = list(1:timedis)):

Creates a list of periods from 1 to the event time or censoring time.

1. unnest(periods):

Expands the list of periods into individual rows.

1. mutate(event = ifelse(periods == timedis & event == 1, 1, 0),

censor = ifelse(periods == timedis & censor == 1, 1, 0)):

Ensures that the event and censor columns are correctly marked only for the period when the event or censoring occurs.

4. filter(periods <= timedis):

Filters out any periods beyond the event or censoring time, ensuring that participants only contribute periods up to the event or censoring time. This should ensure that each individual only contributes periods up to the event or censoring time, and no additional periods are included.

Using this dataset, it is now possible to use the periods as dummy factors and thus estimate intercept parameters for each of the time periods, add a slope parameter for the effect of sex (using female as the reference, or baseline, group), and estimate these parameters using logistic regression whereby the units of analyses are the time periods.

I have called this model “m1” (model 1), and used the ***glm*** function to estimate it:

m1<-glm(event~factor(periods) + male - 1,

family="binomial", data=ppdf)

The key issues of note here are that the ***factor(periods)*** option converts the periods in the dataset into a factor (categorical variable), whereby each unique value in periods will be treated as a separate category and therefore, will have its own parameters. Male is specified as a covariate, while the “-1” is removing estimation of the intercept. The family="binomial" specifies the link function to be used in the generalised linear model.

The results are displayed by invoking:

summary(m1)

A computer screen with numbers and symbols

Description automatically generated

The pattern of the baseline group (females) reflects the general trend we had previously identified, with relatively small differences in the first four periods, a trough in period 5 and a peak in the last period. The slope coefficient for the covariate male is 0.2586, and it is significant at *p* < .05, indicating that being male is associated with a significant upward shift in the logit hazard function of relapse over the study periods.

**Note**: The parameters provided by the logistic regression are in the logit scale. It is possible to transform them into the odds and hazard scales to provide estimates that may be easier to understand or of more interest.

The logit estimates can be transformed into the hazard function estimates using this formula:

Using this formula, we can estimate that, for example, the hazard function estimate of females in the first study period is:

Furthermore, the estimate for the effect of a dichotomous covariate such as male, can be transformed into Odds Ratios (ORs) just by exponentiating the logit estimate. In this example, the Odds Ratios of males relapsing in every time period of the study are:

Therefore, in every period of the study, males are estimated to display approximately 1.30 units increase in the odds of relapsing, compared to females.

Confidence intervals can also be estimated based on the standard errors of the estimated parameter. Considering the parameter for the slope of the male covariates, *βmale* the formula is as follows:

where *ase* indicates the asymptomatic standard error of the estimated parameter.

Considering that in Model 1 (reported above) the estimated = 0.2586 and the = 0.1248, the confidence intervals of the estimated parameter are:

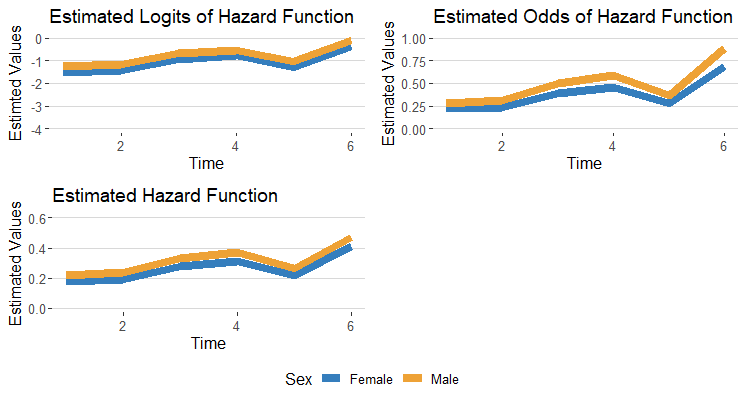
Thus, the 95% Confidence Interval (CI) lies between 0.0140 and 0.5032. Exponentiating these values provides the 95% CI of the Odds Ratios:

;

.

Therefore, in every period of the study, male adolescents are estimated to display a 1.295 increase in the odds of relapsing compared with females, with 95% CI ranging from 1.014 to 1.654.

Transforming the logit estimates into odds and the hazard functions illustrates the predicted risk of the target event estimated based on the logit hazard function model. In the solution to the exercises I carried out these transformations and plotted the fitted lines, as illustrated below.



1. Use the person-period dataset “sa\_exercise\_pp.csv” to fit a model assuming a linear change in the logit hazard function across time periods. Include the covariate “male”. Use a logistic regression to fit the model.

We can consider the “periods” variable akin to a continuous time indicator and model a linear increase in the logits of the hazard function in association with every unit increase in “periods”. To this aim, we can center the indicator of “periods” to a specific period, for example the initial time period:

*#center the periods variable: period 1 is the reference period:*

ppdf$cenperiod<-ppdf$periods-1

We can now fit a linear effect of the time indicator using a logistic regression, and include a slope for the covariate “male”. Note that the model will include the estimate of an intercept:

mlin<-glm(event~cenperiod + male , family="binomial", data=ppdf)

summary(mlin)

A computer screen shot of a black screen

Description automatically generated

The interpretation is analogous to the interpretation of other logistic models. The intercept indicates the expected value of the logit hazard function in the initial period of the study (the interval we chose as the centred time interval) for participants whose covariate value is equal to zero, i.e., females. The logit hazard function is estimated to be -1.537: carrying out the necessary transformations, we estimate the hazard of relapsing for females is approximately 0.18, i.e. females have an estimated 18% probability of relapsing within the first time period of the study, which corresponds to the first two months (The time unit used here is a discrete time period of 2 months, see the description of the dataset.)

The model indicates a significant linear increase in the logit hazard function as a function of the time variable: a one unit increase in the time variable is associated with a 0.203 increase in the logit of the hazard function for females. Using the estimated intercept and the linear time effect, we can calculate the logit hazard functions of female participants in each time period and transform them into estimated hazard functions (i.e., conditional probabilities of the event). The model also indicates a significant increase in the logit hazard function associated with being male: across the time periods, males are estimated to display a 0.282 increase in the logit hazard function of relapsing, compared to females. We can transform this estimate into odds ratios by exponentiating the estimated logit, and calculate *95% CIs*, as displayed in the previous exercise.

Models with polynomials of the time variable can be estimated and compared with the linear model above and the general model of the previous exercise. For example, a quadratic, a cubic, and other polynomials of time can be tested. The different models can then be compared using deviance statistics of information criteria. More details and examples can be found in the references provided with the material for this course.

.