EXPERIMENTAL STUDY OF RECIDIVISM USING SURVIVAL ANALYSIS

Data ScienceTech Institute – Survival Analysis

Authors:
Ali SHEIKHI ,Yassine EL KHATTABI, Hanna ABI AKL,

Date : May 02, 2019

Table of Contents

PROJECT OUTLINE AND DATASET 2
EXPLORATORY DATA ANALYSIS 3
Data Preprocessing and Preliminary Analysis 3
Prepare data for Survival Analysis 4
Kaplan-Mayer and Logrank Test Analysis 5
Kaplan-Mayer 5
LogRank Test 7
MODEL CONSTRUCTION 8
ANoVA - nested model candidates and comparison 8
Comparing non-nested models using the AIC metric 10
Manual Comparison 10
Automatic model selection based on AIC 10
Analysis Based on Residuals 11
Penalized cox regression 13
Creating the `glmnet` model 14
Selecting a threshold through cross-validation for model selection 14
Test the model 15
CCP model 17
EVALUATION OF MODELS 18
CONCLUSION 19

PROJECT OUTLINE AND DATASET

The data is from an experimental study of recidivism of **432** male prisoners, who were observed for a year after being released from prison (Rossi et al., 1980). The interest of this project is to conduct a survival analysis of the released prisoners where the notion of survival is drawn from the number of weeks from time of release until a re-offence. Here, an arrest within the observed year is considered to be an event. Our work begins with exploratory data analysis which will help us in our attempt to model the relationship, if any, between the covariates and the occurrence of an event.

The following variables are included in the data:

- **week** : week of arrest after release, or censoring time
- **arrest** : the event indicator, 1 = arrested , 0 = not
- **fin** : 1 = received financial aid, 0 = not
- **age** : in years at the time of release
- **race** : 1 = black, 0 = others
- **wexp** : 1 = had full-time work experience, 0 = not
- **mar** : 1 = married, 0 = not
- **paro** : 1 = released on parole, 0 = not
- **prio** : number of prior convictions
- **educ** : codes 2 (grade 6 or less), 3 (grades 6 through 9), 4 (grades 10 and 11), 5 (grade 12), or 6 (some post-secondary)
- **emp1 - emp52**: 1 = employed in the corresponding week, 0 = not

EXPLORATORY DATA ANALYSIS

Data Preprocessing and Preliminary Analysis

We load the necessary packages:

```
library(tidyverse)
library(readr)
library(survival)
library(survivalROC)
library(glmnet)
library(survminer)
library(reshape)
```

Looking at the summary of the data we notice that only 26.39% of the subjects get arrested. This corresponds to a low number of events for this dataset and typically we want this number to be higher since it would result in a more representative model, nevertheless we decided to use this dataset as we found the added challenge interesting.

```
dat <- read.csv2("prison.csv", sep = "", header = TRUE)
summary(dat[,1:10])
```

	week	arrest	fin	age	race	wexp	mar	paro	prio	educ
Min.	1.00	0.0000	0.0	17.0	0.0000	0.0000	0.0000	0.0000	0.000	2.000
1st Qu.	50.00	0.0000	0.0	20.0	1.0000	0.0000	0.0000	0.0000	1.000	3.000
Median	52.00	0.0000	0.5	23.0	1.0000	1.0000	0.0000	1.0000	2.000	3.000
Mean	45.85	0.2639	0.5	24.6	0.8773	0.5718	0.1227	0.6181	2.984	3.477
3rd Qu.	52.00	1.0000	1.0	27.0	1.0000	1.0000	0.0000	1.0000	4.000	4.000
Max.	52.00	1.0000	1.0	44.0	1.0000	1.0000	1.0000	1.0000	18.000	6.000

A first look at the dataset (excluding cols emp1-52):

```
head(dat[,1:10])
```

	week	arrest	fin	age	race	wexp	mar	paro	prio	educ
1	20	1	0	27	1	0	0	1	3	3
2	17	1	0	18	1	0	0	1	8	4
3	25	1	0	19	0	1	0	1	13	3
4	52	0	1	23	1	1	1	1	1	5
5	52	0	0	19	0	1	0	1	3	3
6	52	0	0	24	1	1	0	0	2	4

Variables corresponding to `financial aid`, `race`, `work experience`, `married` and `parole` are **categorical** and need to be defined as factors in R:

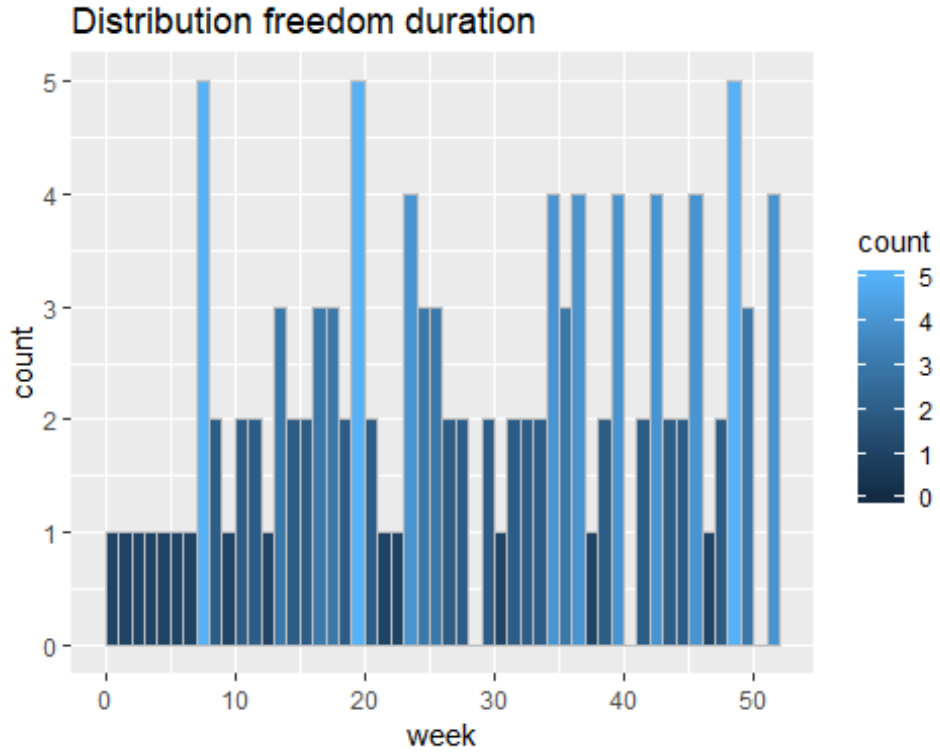
```
df <- mutate(dat,
   fin   = factor(fin,  levels = c('1', '0'), labels = c('financed', 'not_financed')),
   race  = factor(race, levels = c('1', '0'), labels = c('black', 'other')),
   wexp  = factor(wexp, levels = c('1', '0'), labels =c('full_time','not_full_time')),
   mar   = factor(mar,  levels = c('1', '0'), labels = c('married', 'single')),
   paro  = factor(paro, levels = c('1', '0'), labels = c('parole', 'not_parole'))
)
```

As mentioned earlier, **26.39%** of the subjects are arrested within the year and those that weren't arrested made it through the whole year. From the following we can confirm that there are no individuals that were not arrested and for some reason didn't reach the end of the year.

```
prop.table(table(df$arrest != 1))
     FALSE      TRUE
 0.2638889 0.7361111

prop.table(table(df$week <= 52 & df$arrest != 1))
     FALSE      TRUE
 0.2638889 0.7361111
```

In order to get an idea of the distribution of 'time to event' we can plot the histogram for the `week` column for those that were arrested. Most subjects reach week 52 and so we have omitted this from the histogram to get a clearer picture. There isn't much that can be said except that for the first 7 weeks there is a visible effort to stay out of trouble.

```
ggplot(df[df$arrest == 1,], aes(x=week)) + geom_histogram(breaks=seq(0, 52, by =1), col="grey",
aes(fill=..count..))  + ggtitle("Distribution freedom duration")
```



Prepare data for Survival Analysis

Survival analysis in R requires that the target or the response composite variable be defined as a 'Surv' object. Thus, we create the Surv object 'y' from columns `week` and `arrest`. The covariates being considered do not include columns emp1-52. We suspect that information contained within emp1-52, such as 'weeks until arrest' or 'week prior to arrest employment status', will significantly improve the final model, unfortunately we cannot use this since we shall not avail the model of these columns in the test set.

```
y <- Surv(df$week, df$arrest)
x <- df[,3:10]
```

The data is split randomly into a training and a testing set (80% / 20%)

```
set.seed(1234)

train.size <- 0.8

i.training <- sample(nrow(x), size = as.integer(432*train.size), replace = FALSE)
i.testing <- setdiff(seq_len(nrow(x)), i.training)

x.training <- x[i.training,, drop = FALSE]
y.training <- y[i.training,, drop = FALSE]

x.testing <- x[i.testing,, drop = FALSE]
y.testing <- y[i.testing,, drop = FALSE]
```

Kaplan-Mayer and Logrank Test Analysis

Kaplan-Mayer

Kaplan-Meier is a powerful non-parametric method to estimate the survival curve. A survfit object is created from the number of individuals at risk and number of events at each possible arrest time.

```
fit.KM <- survfit(y.training ~ 1, data = x.training)
fit.KM

## Call: survfit(formula = y.training ~ 1, data = x.training)
##
##       n  events  median 0.95LCL 0.95UCL
##     345      90      NA      NA      NA
```

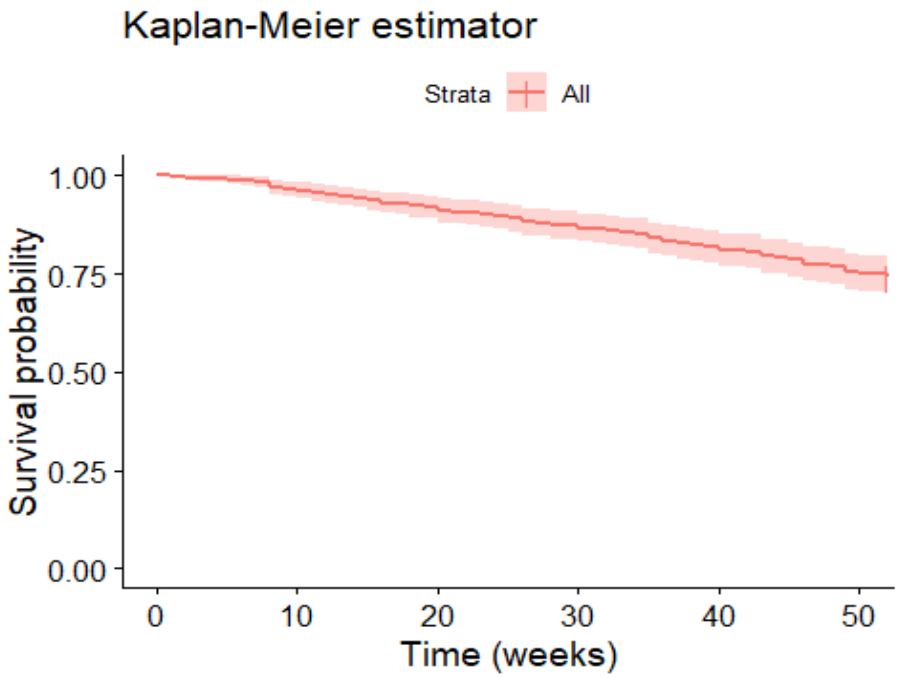
Note that the median is NA. This is to be expected because the survival at the end of the experiment is 74% so we never encounter the 50% survival rate during the experiment time.

Another thing to notice is that the ratio of the arrests (90/345 = 0.26) is still 26% after the train-test split. This confirms that the split has been done correctly.

Next we plot the survival curve. It can be seen that the survival rate, or in this case the probability of not getting arrested, decreases over time in an almost linear fashion.

Note that, as expected, the ratio of survival at the end of the experiment is **0.74**. Using **censor** parameter gives a single mark at the end of the experiment, which is logical given that all the censoring is occurs at week 52.

```
ggsurvplot(fit.KM, data = x.training, title = "Kaplan-Meier estimator",
     ylab = "Survival probability",
     xlab = "Time (weeks)",
     censor.shape="|", censor.size = 4)
```

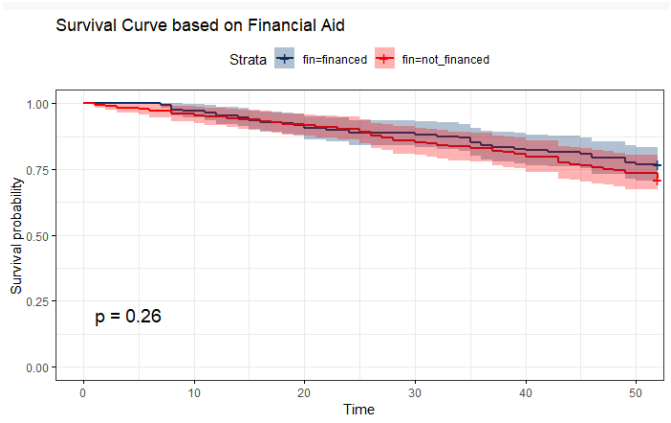


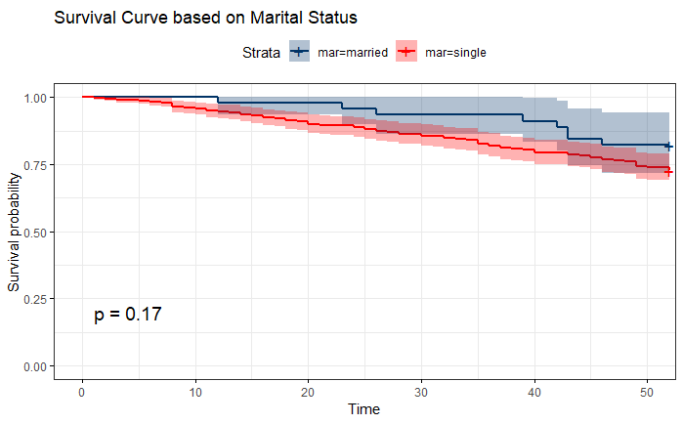

```
fit.KM

## Call: survfit(formula = y.training ~ 1, data = x.training)
##
##       n  events  median 0.95LCL 0.95UCL
##     345      90      NA      NA      NA
```

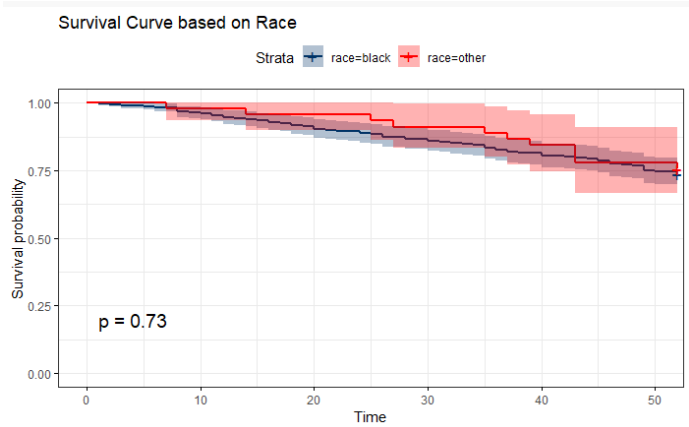
The next step is to plot the effects of the different categorical covariates on the survival curve.

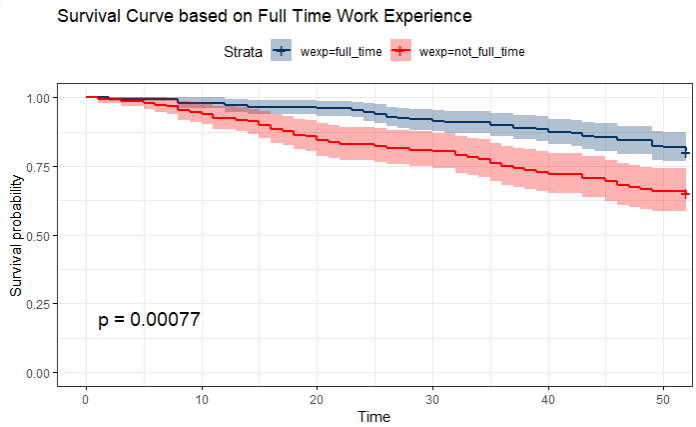
The following are plots for the 5 categorical variables. The code for the plots is the same with minor differences. As an example, we have included this code for 'Financial Aid' below.

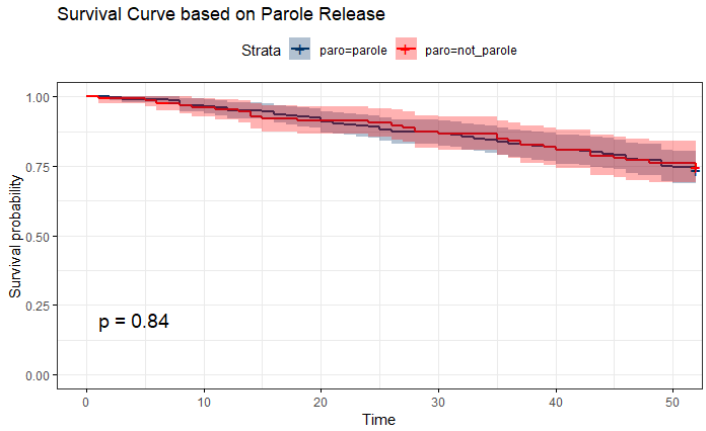
```
fit.KMfin <- survfit(y.training ~ fin, data = x.training)

ggsurvplot(
  fit.KMfin,
  title = "Survival Curve based on Financial Aid",
  size = 1,                 # change line size
  palette =
    c("#003366", "#FF0000"),# custom color palettes
  conf.int = TRUE,          # Add confidence interval
  pval = TRUE,              # Add p-value
    ggtheme = theme_bw()      # Change ggplot2 theme
)
```











Given the Kaplan-Meier plots, it seems that only `Work Experience` may have an effect on the survival curve.

LogRank Test

We can investigate further with the logrank test for each of the 5 categorical (binary) covariates:

```
fit.logrankfin  <- survdiff(y.training ~ fin, data = x.training)
fit.logrankmar  <- survdiff(y.training ~ mar, data = x.training)
fit.logrankrace <- survdiff(y.training ~ race, data = x.training)
fit.logrankwexp <- survdiff(y.training ~ wexp, data = x.training)
fit.logrankparo <- survdiff(y.training ~ paro, data = x.training)
```

Table of results:

Covariate	Samples	Observed	Expected	P-value
fin (financed | non-financed)	168 | 177	39 | 51	44.3 | 45.7	0.3
mar (married | single)	45 | 300	8 | 82	12.5 | 77.5	0.2
race (black | other)	300 | 45	79 | 11	77.9 | 12.1	0.7
wexp (full time | none)	198 | 14	39 | 51	54.5 | 35.5	8e-04
paro (parole | none)	219 | 12	58 | 32	57.1 | 32.9	0.8

wexp is the only one that has a significant effect on the survival rate judging by the p-value of the test.

MODEL CONSTRUCTION

In this section, we are going to follow four techniques to build a model :
(I) Anova, (II) AIC, (III) Elastic net and (IV) CCP

ANoVA - nested model candidates and comparison

The purpose here is to find the best model with the minimum complexity that describes and predicts the occurrence of the event in this study. Anova offers a way to compare nested models and thus the model of interest shall be the one with minimum covariates without significant loss in performance (Note: Due to the high number of comparisons involved in a comprehensive analysis, we have restricted our final model of interest to one with just 2 covariates)

In this section we create the following models:

- **M.total** is the global model that contains all the covariates
- **M.0** is the model with no covariate (the assumption is that there is no link between the event occurrence and the covariates that we have)
- **M.prio**, **M.age**, **M.mar**, **M.fin**, **M.wexp** and **M.educ** are models containing one covariate each, as indicated by the model names, that are going to be compared to the **M.0** and **M.total** models (Note: Covariates corresponding to race and release on parole were not considered due to very high p-values in both Kaplan-Meier and LogRank tests)

```
M.0     <- coxph( y.training ~ 1    , data = x.training )
M.prio  <- coxph( y.training ~ prio , data = x.training )
M.age   <- coxph( y.training ~ age  , data = x.training )
M.mar   <- coxph( y.training ~ mar  , data = x.training )
M.fin   <- coxph( y.training ~ fin  , data = x.training )
M.wexp  <- coxph( y.training ~ wexp , data = x.training )
M.educ  <- coxph( y.training ~ educ , data = x.training )
M.total <- coxph( y.training ~ fin + age + race + wexp + mar + paro + prio + educ,data =
x.training)
```

After creating the models, let's start by trying to identify which of our single-covariate model is the best in comparison to the empty model:

Model	P-value	Log-Likelihood Difference
anova(M.0, M.prio)	0.001295 **	5.17
anova(M.0, M.age)	0.0003444 ***	6.4
anova(M.0, M.mar)	0.1414	1.08
anova(M.0, M.fin)	0.2587	0.64
anova(M.0, M.wexp)	0.0009459 ***	5.46
anova(M.0, M.educ)	0.03356 *	2.26

Analyzing the results above, we can say that if we have to choose only one covariate, the most significant one would be age in comparison to the empty model.

Let's now compare M.age to a model with age + another covariate. For that we need to create the following models:

```
M.age.prio  <- coxph(y.training ~ age + prio, data = x.training)
M.age.mar   <- coxph(y.training ~ age + mar , data = x.training)
M.age.fin   <- coxph(y.training ~ age + fin , data = x.training)
M.age.wexp  <- coxph(y.training ~ age + wexp , data = x.training)
M.age.educ  <- coxph(y.training ~ age + educ , data = x.training)
```

Now let's see which of these is best in comparison to M.age:

Model	P-value	Log-Likelihood Difference
anova(M.age , M.age.prio)	0.003997 **	4.15
anova(M.age , M.age.mar)	0.4717	0.26
anova(M.age , M.age.fin)	0.3897	0.37
anova(M.age , M.age.wexp)	0.03933 *	2.13
anova(M.age , M.age.educ)	0.05967	1.78

From the results above, we can see that the best model with 2 covariates in comparison to M.age is M.age.prio

Let's see how the model M.age.prio is doing in comparison to M.total

```
anova(M.age.prio  , M.total)

 Analysis of Deviance Table
  Cox model: response is  y.training
  Model 1: ~ age + prio
  Model 2: ~ fin + age + race + wexp + mar + paro + prio + educ
    loglik  Chisq Df P(>|Chi|)
 1 -502.60
 2 -500.04 5.1202  6    0.5285
```

The difference is not significant (i.e. the p-value is big, and the loglik difference is not that important).

In conclusion, using the nested model comparison with ANoVA, the best model is M.age.prio.

Comparing non-nested models using the AIC metric

Manual Comparison

AIC is a metric that enable us to compare different models even if they are not nested.

```
fits <- list(M.prio = M.prio, M.age = M.age, M.mar = M.mar, M.wexp = M.wexp , M.educ = M.educ,
             M.age.fin = M.age.fin ,M.age.prio = M.age.prio, M.age.wexp = M.age.wexp, M.age.educ =
M.age.educ,
             M.age.mar = M.age.mar, M.total = M.total)
sapply(fits, AIC)

M.prio       M.age        M.mar        M.wexp    M.educ    M.age.fin    M.age.prio
1017.954     1015.491     1026.141     1017.373  1023.787  1016.752     1009.206
M.age.wexp   M.age.educ   M.age.mar    M.total
1013.245     1013.945     1016.973     1016.086
```

As suspected, the best AIC model is `M.age.prio`.

Automatic model selection based on AIC

Alternatively, and more conveniently, we can use the automatic selection using the function `step`:

```
MAIC <- step(M.total)
```

In the interest of reporting in a concise manner, the following table summarizes the output of the step function.

Step	Model Covariates	AIC
0	fin + age + race + wexp + mar + paro + prio + educ	1016.09
1	fin + age + race + wexp + mar + prio + educ	1014.20
2	fin + age + race + wexp + prio + educ	1012.42
3	fin + age + wexp + prio + educ	1010.98
4	age + wexp + prio + educ	1009.63
5	age + wexp + prio	1009.22
6	age + prio	1009.21

Once again, the model selected at the end is the one related to `age` and `prio`.

```
MAIC

      Call:
      coxph(formula = y.training ~ age + prio, data = x.training)

         coef exp(coef) se(coef)     z       p
  age  -0.07083   0.93162  0.02393 -2.96 0.00307
  prio  0.09520   1.09988  0.03041  3.13 0.00175
  Likelihood ratio test=21.1  on 2 df, p=2.623e-05
  n= 345, number of events= 90
```

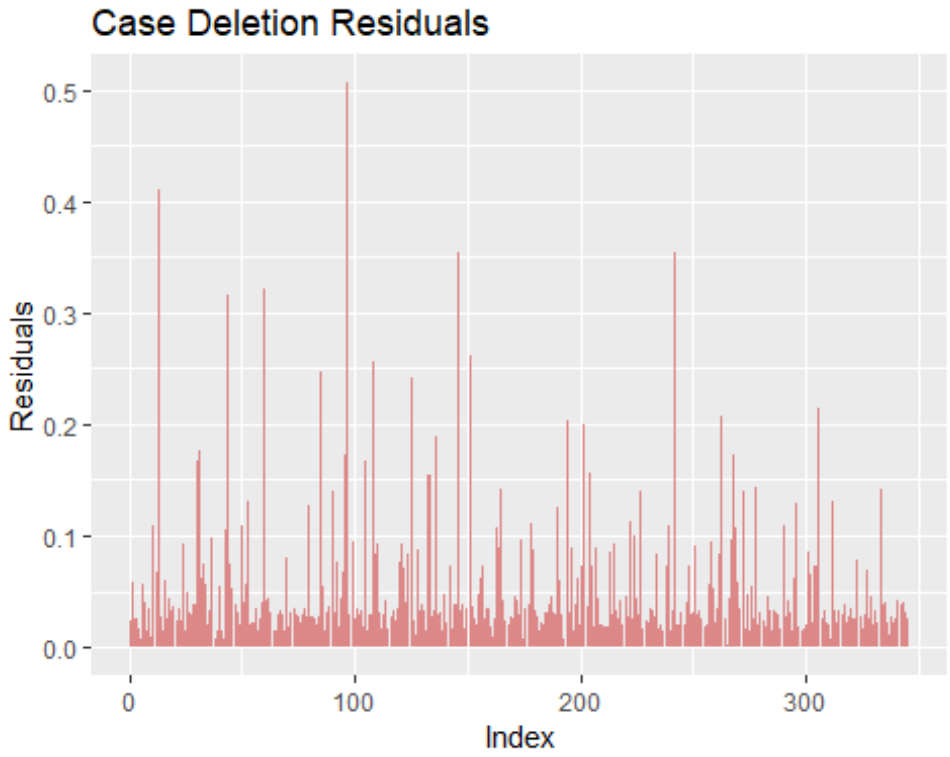
Analysis Based on Residuals

In the following part, we will try to illustrate the different uses of residuals to either identify any potential outliers, or any missing functional form of the covariates. We are going to use case deletion residuals, Martingale residuals and Schoenfeld residuals.

Case Deletion Residuals

The purpose of this analysis is to understand the effect of each sample on the values of the model coefficients (or betas) pertaining to `MAIC`. If we have an extreme value for a given sample, that means it has a big influence on the coefficients and can be considered as an outlier.

```
res <- data.frame(matrix(ncol = 1, nrow = as.integer(432*train.size)))

dfbetas <- residuals(MAIC, type = 'dfbetas')
res$MAICcdr <- sqrt(rowSums(dfbetas^2))

ggplot(aes(x = as.numeric(row.names(res)), y = MAICcdr), data = res) +
  geom_bar(fill="#DD8888",stat = 'identity') +
  labs(x='Index', y='Residuals') +
  ggtitle("Case Deletion Residuals")
```



Here we do not see any outliers. We can say though that we have two kinds of samples: the one with big values for residuals and the one with small values. Our guess is that the big values are related to the samples experiencing an event of arrest. These are the source of "information" for the model. Since in our dataset only few samples experience the event **arrest** (about 26%), it is logical that they appear with significant values of residuals because if we suppress any of them we lose a significant amount of information.

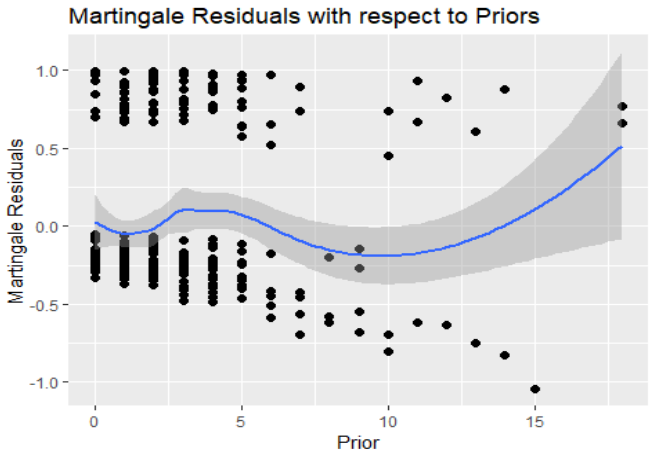
Martingale Residuals

In order to have a better idea about how good our model fits the data, i.e. `age` and `prio` explains sufficiently the risk, we can compute Martingale residuals for our model named `MAIC`.

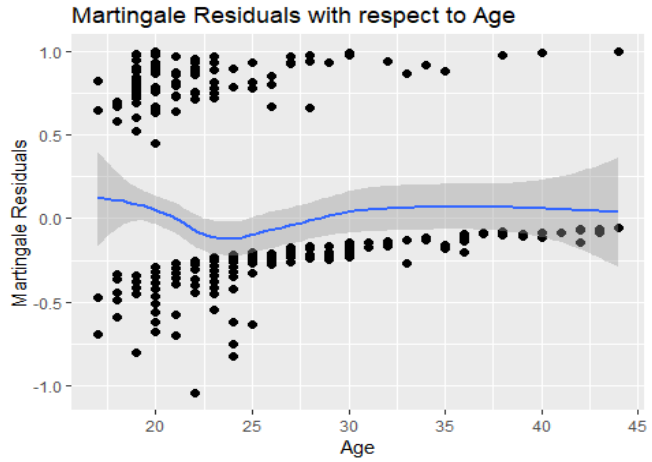
```
res$MAICmartins <- residuals(MAIC, type = "martingale")
```

Let us now plot the residuals

```
par(mfrow = c(1, 2), mar = c(4.2, 2, 2, 2))

ggplot(data=x.training,aes(x=prio, y=res$MAICmartins)) +
  geom_point(size=2) + geom_smooth() +
  ggtitle("Martingale Residuals with respect to Priors") +
  labs(x='Prior', y='Martingale Residuals')
```




```
ggplot(data=x.training,aes(x=age, y=res$MAICmartins)) +
  geom_point(size=2) + geom_smooth() +
  ggtitle("Martingale Residuals with respect to Age") +
  labs(x='Age', y='Martingale Residuals')
```



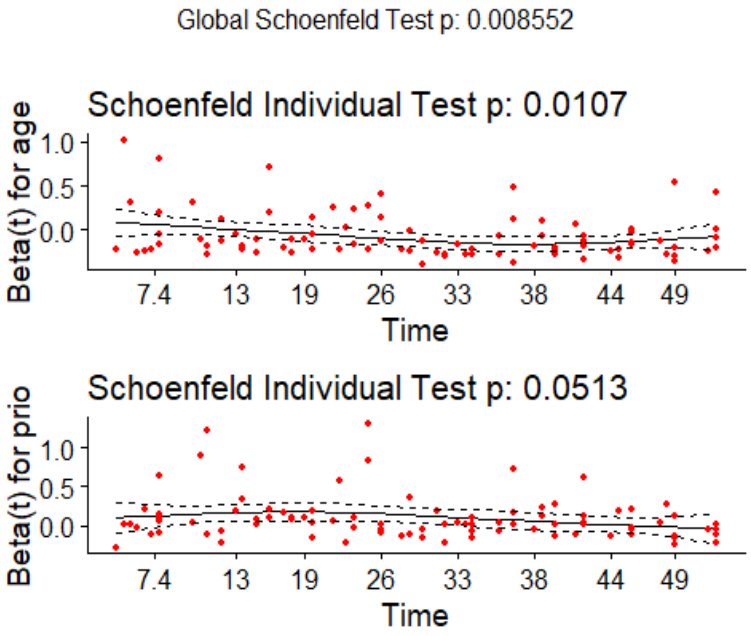
From the cloud of points in the plots of residuals against age and prio, the following remarks can be drawn:

- There is no real trend that suggests adding another form of the covariate (we can debate that for `prio`, the form of the line is quadratic, but we do not consider it as a strong trend)
- There appears to be 2 clusters of similar number of points in the 2 plots. A suggestion might be to conduct stratified analysis by a binary covariate. Since the number of points in both clusters seem to be similar, the stratification covariates that we can consider are either `wexp` (work experience) or `fin` (financial aids)

Schoenfeld Residuals

In this part, we want to test if the Hazard (i.e. risk of being arrested) in relation to the time, is proportional to the covariate we selected. For this we use `cox.zph()` function applied to our model `MAIC`:

```
residual.sch <- cox.zph(MAIC)
residual.sch

##           rho chisq       p
## age    -0.230  6.51 0.01074
## prio   -0.203  3.80 0.05127
## GLOBAL     NA  9.52 0.00855

ggcoxzph(residual.sch, resid = TRUE, se = TRUE, df = 4, nsmo = 40, var=c("age", "prio"),
  point.col = "red", point.size = 1, point.shape = 19, point.alpha = 1,
  caption = NULL, ggtheme = theme_survminer())
```



In the plots, thanks to the smoothed line, we can see that the effect of `age` and `prio` is decreasing with time. The P-value in cox.zph test suggest that this is true in particular for `age`.

Penalized cox regression

In the previous sections, we tried to build a model manually. We are starting by using Kaplan-Mayer and Logrank test to have an idea about the most significant covariates, then using Anova we compared nested models and we found out that `prio` and `age` are the most significant covariates.

We obtained the same results using the AIC criterion for an automatic model selection thanks to `step` function.

We started by the model with all covariates `M.full` and the reverse step enabled us to have a final model with the covariates `age` and `prio`.

For this part, we are going to use `glmnet` in order to be able take advantage of penalization and cross validation to select the model with the best tradeoff between good fitness and complexity.

Creating the `glmnet` model

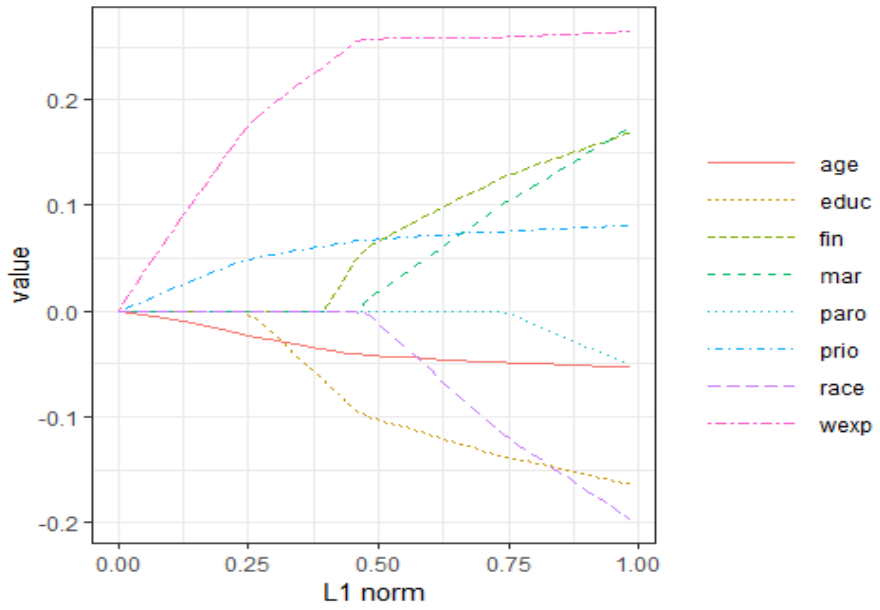
Data preprocessing and model fitting

To be able to use glmnet, we need to format our data as a matrix. We will do this for `x.training` and `x.testing`:

```
xmat.training <-  data.matrix(x.training, rownames.force = NA)
xmat.testing <-  data.matrix(x.testing, rownames.force = NA)
```

Let's fit the model using `cox` family:

```
M.glm <- glmnet(xmat.training, y.training, family = "cox")
beta=coef(M.glm)

tmp          <- as.data.frame(as.matrix(beta))
tmp$coef     <- row.names(tmp)
tmp          <- reshape::melt(tmp, id = "coef")
tmp$variable <- as.numeric(gsub("s", "", tmp$variable))
tmp$lambda   <- M.glm$lambda[tmp$variable+1] # extract the lambda values
tmp$norm     <- apply(abs(beta[-1,]), 2, sum)[tmp$variable+1] # compute L1 norm


ggplot(tmp[tmp$coef != "(Intercept)",], aes(norm, value, color = coef, linetype = coef)) +
    geom_line() +
    xlab("L1 norm") +
    guides(color = guide_legend(title = ""),
           linetype = guide_legend(title = "")) +
    theme_bw() +
    theme(legend.key.width = unit(3,"lines"))
```

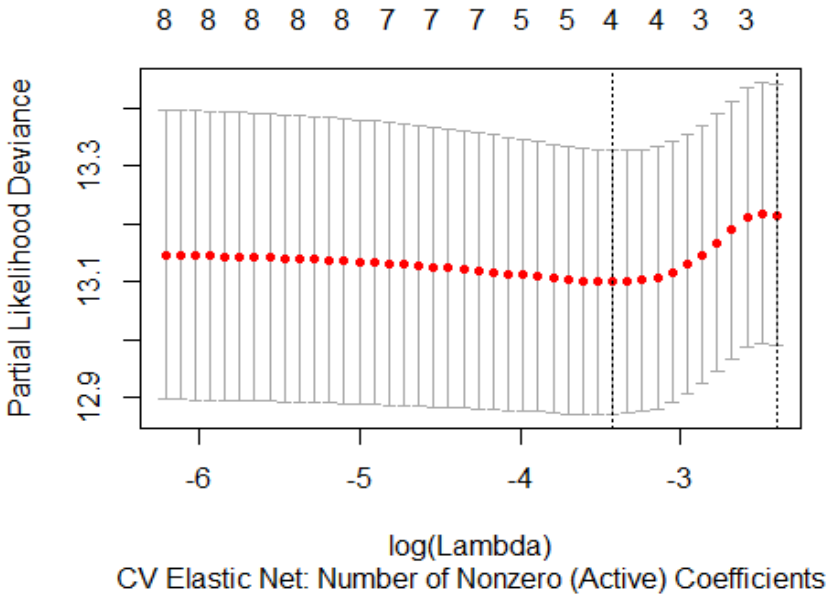


The plot above explains the effect of penalization, in this case an elastic combination of Ridge and Lasso by default, on the number of selected covariates.

Selecting a threshold through cross-validation for model selection

The `glmnet` function allows us to identify the best model using as its metric the `Partial Likelihood Deviance.`

```
set.seed(1234)
M.cv10 <- cv.glmnet(xmat.training, y.training, family = "cox")
plot(M.cv10,  sub="\nCV Elastic Net: Number of Nonzero (Active) Coefficients")
```



From the plot, the minimum error is obtained with the model using 4 covariates. Here we cannot select the model with `lambda.1se` error because it is higher than the maximal error.

Let's retrieve the coefficients of the model with `lambda.min` error instead.

```
b <- coef(M.cv10, s = "lambda.min")
b.enet <- b[b!=0]

## <sparse>[ <logic> ] : .M.sub.i.logical() maybe inefficient

names(b.enet) <- colnames(x)[as.logical(b != 0)]
b.enet

##         age        wexp        prio        educ
## -0.03143891  0.21373034  0.05731713 -0.04311371
```

The covariates selected by `cv.glmnet` are `prio`, `age`, `wexp` and `educ`. This is consistent with the findings of the previously used methods, in particular the `MAIC`.

Test the model

Later in the study we are going to make predictions with all the constructed models. But here, fore the sake of clarifications, we are going to detail the testing part by illustrating how to make predictions and how to assess their quality. We constructed the model thanks to the training set, and now we are going to conduct the tests on the testing set. We should mention here that, in the sense of survival analysis, predictions are a score which is related to the hazard. The snippet of code bellow allows us to make predictions (using the model with the smallest partial likelihood deviance)

```
score.testing <- predict(M.cv10, newx = xmat.testing, s = "lambda.min")
score.testing <- score.testing / IQR(score.testing)
```

Here we normalize the predicted score as a good practice

To be able to assess the quality of hazard prediction by our model, we can use cox regression to measure the proportionality of the scores we predicted in relation to the surv object of our testing data.

```
summary(coxph(y.testing ~ score.testing))
```

```
## Call:
## coxph(formula = y.testing ~ score.testing)
##
##   n= 87, number of events= 24
##
##                 coef exp(coef) se(coef)     z Pr(>|z|)
## score.testing 0.5183    1.6791   0.2749 1.886   0.0594 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##               exp(coef) exp(-coef) lower .95 upper .95
## score.testing     1.679     0.5956    0.9798     2.878
##
## Concordance= 0.596  (se = 0.061 )
## Likelihood ratio test= 3.65  on 1 df,   p=0.06
## Wald test            = 3.56  on 1 df,   p=0.06
## Score (logrank) test = 3.57  on 1 df,   p=0.06
```

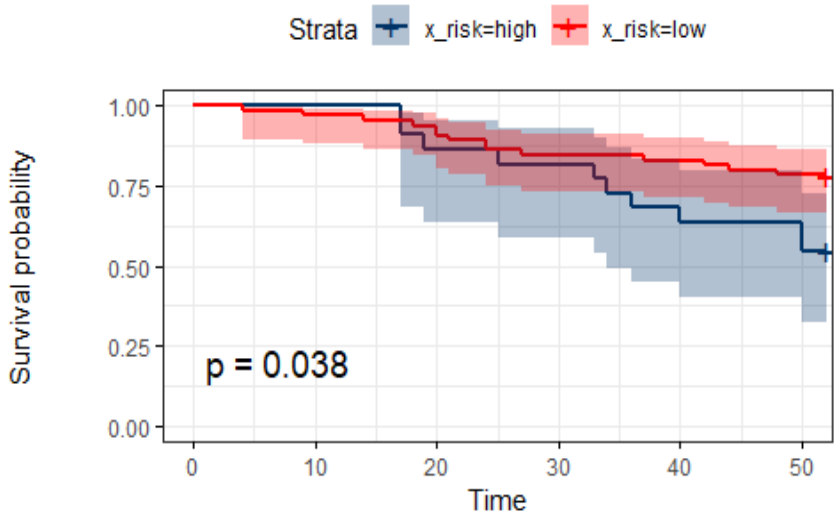
The P-value of the test is close to 5% (slightly higher) but we can consider it as a positive sign for the quality of the predictions on the testing set.

Furthermore, in order to be able to use logrank test, we can create a categorical variable using a cut-off threshold on our predictions. Here we chose the 3rd quartile as a cut-off knowing that 26% of our samples have experienced the event arrest.

```
x_risk <- ifelse(score.testing <= quantile(score.testing)[4], "low", "high")
table(x_risk)
```

```
## x_risk
## high  low
##   22   65
```

```
fit.KM.x_risk <- survfit(y.testing ~ x_risk, conf.type = "log-log")
ggsurvplot(
  fit.KM.x_risk,
  data = x.testing,
  size = 1,                 # change line size
  palette =
    c("#003366", "#FF0000"),# custom color palettes
  conf.int = TRUE,          # Add confidence interval
  pval = TRUE,              # Add p-value
  risk.table = TRUE,        # Add risk table
  risk.table.col = "strata",# Risk table color by groups
      # Change legend labels
  risk.table.height = 0.25, # Useful to change when you have multiple groups
  ggtheme = theme_bw()      # Change ggplot2 theme
)
```



In assessing the effect of the binary encoded transformation of our predictions (i.e. “high” and “low”) on the survival curve of the testing data we find the P-Value to be significant.

```
LR <- survdiff(y.testing ~ x_risk)
LR

## Call:
## survdiff(formula = y.testing ~ x_risk)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## x_risk=high 22       10     5.69      3.27      4.31
## x_risk=low  65       14    18.31      1.02      4.31
##
##  Chisq= 4.3  on 1 degrees of freedom, p= 0.04
```

From the logrank test, we see that the p-value is significant, this is also a good sign that the split we made makes sense and the model made reasonable predictions.

CCP model

As a final model, we are creating the CCP model which is basically just stacking the estimated coxph coefficients computed for each single covariate in their own respective models. So, we build this model taking in account all the covariates.

```
fits <- plyr::adply(xmat.training, 2, function(x) broom::tidy(coxph(y.training ~ x)))
b.CCP <- with(fits, structure(estimate, names = as.character(X1)))
```

EVALUATION OF MODELS

Here we retrieve the coefficients of all the models of interest previously described to make predictions on the testing set:

```
b.total <- coef(M.total)
b.MAIC      <- coef(MAIC)

names(b.MAIC) = c("age","prio")
names(b.total)  = c("fin", "age", "race", "wexp" ,"mar", "paro", "prio", "educ")
models_coefficients <- tibble(
  method = c("manual", "Anova-aic", "elasticNet", "ccp"),
  coefficients = list(b.total, b.MAIC, b.enet, b.CCP)
)
```

The following code is creating a function that makes the predictions through a linear combination of each model's coefficients with the values of covariates for each sample.

```
lincom <- function(b, X) rowSums(sweep(X[, names(b), drop = FALSE], 2, b, FUN = "*"))
```

Let's predict and compare the quality of predictions using the cox regression between the ground truth surv object and the standardized predicted score related to the hazard.

```
models_performance <- mutate(models_coefficients,
                             predictions = map(coefficients, ~ lincom(., xmat.testing)),
                             cox_obj = map(predictions, ~ coxph(y.testing ~ I(. / sd(.)))),
                             cox_tab = map(cox_obj, broom::tidy)
) %>%
  unnest(cox_tab)
models_performance <- mutate(models_performance,
                             AUC = map_dbl(predictions, ~ survivalROC::survivalROC(y.testing[, 1],
y.testing[, 2], ., predict.time = 52, method = "KM")$AUC)
) %>%
  select(method, estimate, std.error, p.value, AUC)
```

Model	Estimate	P-value	AUC
Manual	0.5910676	0.01411791	0.6646825
Anova	0.4897727	0.04257515	0.6352513
Elasticnet	0.4067599	0.05935179	0.6250000
CCP	0.5276741	0.02458979	0.6567460

From the table below, looking at the P-Value, then the estimate and AUC, the best model seems to be the total model. This is not surprising giving the fact that we used all the covariates.

The second-best model is CCP, which indicates that the approach gives some descent results with minimal effort. Comparing the rest of the models, we can see that `Anova-aic` (named as such since both Anova and MAIC resulted in the same 2 covariates) is better than `elasticNet` with less complexity (2 coefficients instead of 4).

CONCLUSION

Initially, preliminary insights into the potential significance of the existing categorical covariates in our dataset were obtained through the Kaplan-Meier and Logrank tests. In this case, we found full-time work experience (`wexp`) to be the only one determined as important by both tests.

Next, a carefully designed series of comparisons of nested models using ANoVA further strengthened and shaped our assumptions. The resulting best 2-covariate model from this procedure had as predictor variables `prio` and `age`.

Moreover, the significance of these two covariates was confirmed by the subsequent AIC-metric based `MAIC` model using automatic reverse-step covariate selection, since the final model had precisely the same two variables. Our `glmnet` model with complexity regularization facilitated by Lasso-Ridge penalty consisted of four covariates, namely, `prio`, `age`, `wexp` and `educ`. These were consistent with our previous findings, with the exception of `educ`.

Finally, after creating the `CCP` model, and using the full model as a reference we compare and evaluate all of them. If we are looking for a fairly simple and yet robust model, our choice would be M.age.prio (the model using age and prio and predictors).