Survival Analysis of Game of Thrones Characters

ABSTRACT

The hit HBO drama series Game of Thrones has become notorious for its shocking and ruthless killings of major characters throughout its run. In this whitepaper I try to identify the main risk covariates that would predict survival using Kaplan-Meier estimators, Cox regression, Elastic Net and stepwise methods for variable selection

CRIMI, Mariano

March 3rd, 2020

1 INTRODUCTION

Game of Thrones is a hit fantasy tv show based on the equally famous book series "A Song of Fire and Ice" by George RR Martin. The show is well known for its vastly complicated political landscape, large number of characters, and its frequent character deaths.

In this analysis we'll use a dataset containing right censored data on survival for **359 characters** of the show series along with some data of their death, allegiance, sex, religion, occupation, social status, etc.

The intention of this study is to analyze survival for different groups and develop a model that can accurately predict death risk for a given set of covariates.

1.1 RESOURCES

Dataset: https://github.com/mcrimi/got_survival/blob/master/character_data_S01-S08.csv

Data dictionary: https://github.com/mcrimi/got_survival/blob/master/data_dictionary.pdf

R Notebook: https://github.com/mcrimi/got_survival/blob/master/Paper.Rmd

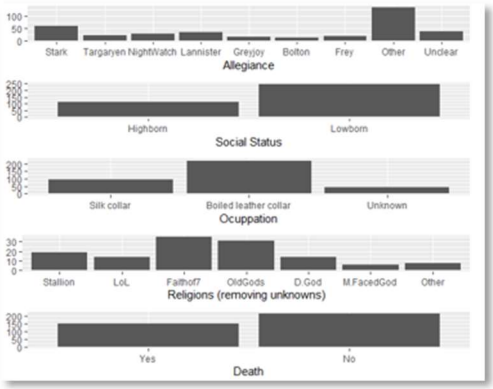
2 DATA WRANGLING

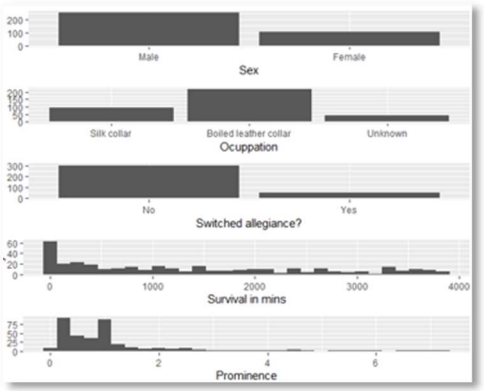
Analysis was preceded by a process of cleaning the dataset and selecting the explanatory variables of interest. All the metadata of the deaths circumstances, cause, places, etc. as well as names of characters and id's were removed as they're irrelevant to the analysis. I've also decided not to consider all the data related to the introduction of the characters (intro_season, intro_time, intro_episode, etc) as in my view are considered as confounding factors and would introduce a kind of censoring that is not part of this analysis. Additionally I've converted all categorical data into factors in my data frame according to the levels and labels provided in the data dictionary.

3 BASIC EXPLORATION OF THE DATA

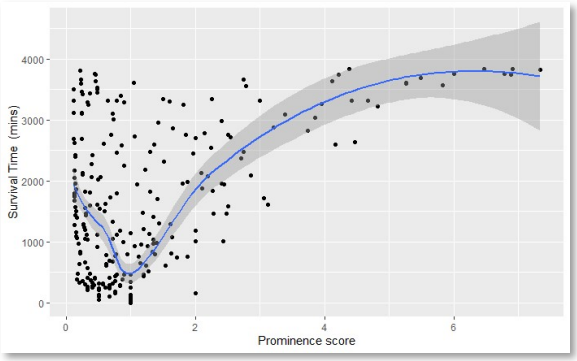
After data curation and cleaning we end up with a dataset containing **359 records (n)** for **7 potential features (p)**, of which only one is continuous and the other seven are categorical. Upon exploration we find no missing values in the dataset, so no imputation is need. We find that we are in the presence of a male dominated show in terms of characters (254 males vs 105 females). Religions are normally distributed (for the characters we know their religion). A large majority has stayed faithful to their allegiance during the show (only 15% of the characters switched allegiance) and an astonishing 40% of the characters had died by the end of the experiment (n=147)

social_status	allegiance_switched	sex
Highborn:112	No :304	Male :254
Lowborn :247	Yes: 55	Female:105
allegiance_last	**prominence**	**religion**
Other :135	Min. :0.1111	Unknown :233
Stark : 60	1st Qu.:0.3333	Faithof7: 35
Unclear : 37	Median :0.8750	OldGods : 31
Lannister : 34	Mean :1.1292	Stallion: 19
NightWatch: 27	3rd Qu.:1.1716	LoL : 14
Targaryen : 21	Max. :7.3425	D.God : 14
(Other) : 45		(Other) : 13
exp_time_sec	**occupation**	**dth_flag**
Min. : 8	Silk collar :96	Min. :0.0000
1st Qu.: 14496	Boiled leather collar:221	1st Qu.:0.0000
Median : 66551	Unknown :42	Median :1.0000
Mean : 81644		Mean :0.5905
3rd Qu.:144592		3rd Qu.:1.0000
Max. :230347		Max. :1.0000





We see prominence's distribution resembling an exponential distribution of some sort. Prominence is a strange beast. According to the data dictionary this metric is calculated as follows: [prominence] = ([featured_episode_count] / [exp_episode]) * [exp_season] so it's basically a ratio of appearance over lifetime, weighted by season survival. Given it is a function of the survival we would expect to see some correlation with survival time. When we do a rudimentary exploration of the relation between prominence score and survival time we see that the relation doesn't seem to be lineal.



Based on this I've decided to convert this continuous variable into a factor. Declaring a score below 1 as low prominence, 1 to 4 medium and 4-9 as high prominence.

```
dat$prominence<- cut(dat$prominence, breaks=c(0,1,4,9), label=c("low","med","high"))
```

This leaves us with an even sharper exponential distribution of the counts and exclusively categorical covariates for our analysis.

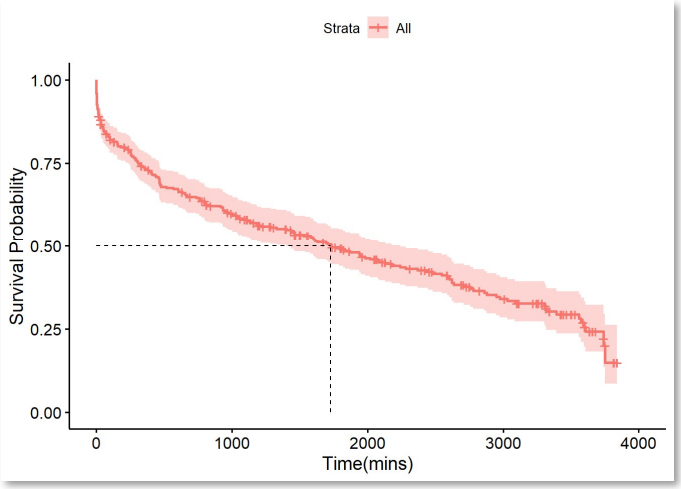
After data wrangling and cleaning we can then attach a survival object to be used as the response variable of our dataset using `survival::Surv`. Object Y will consist on Time (right censored) **in minutes** and Event (0=alive, 1=dead).

```
dat$y <- with(dat, Surv(time= exp_time_sec/60, event=dth_flag))
```

4 SURVIVAL ANALYSIS

In this section we proceed to explore the general survival Kaplan-Meier curve for our characters survival:

```
fitKM<-survfit(y ~ 1, data = dat)
ggsurvplot(fitKM, xlab="Time(mins)", ylab="Survival Probability",main="KM Plot", surv.median.lin="hv")
```

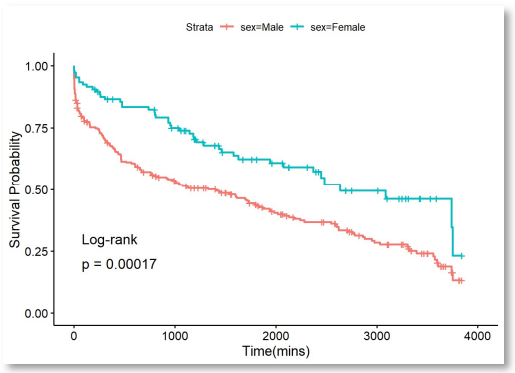


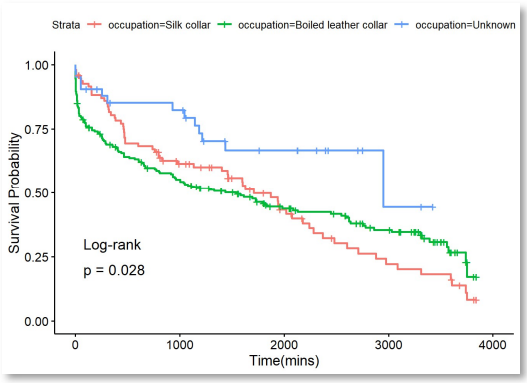

```
## Call: survfit(formula = y ~ 1, data = dat)
##
##       n  events  median 0.95LCL 0.95UCL
##     359     212    1729    1403    2242
```

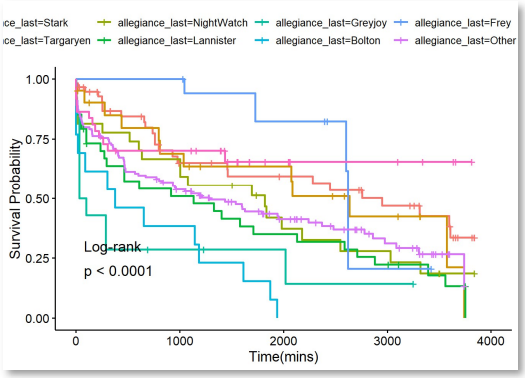
We see a **median survival of 1729 minutes** (with 95% confidence that the actual median is within 1403 mins and 2242 mins) with the first death event happening as soon as 8 seconds into their appearance on the show and some characters surviving as much as 62 hours and 30 mins before their (most surely violent) death.

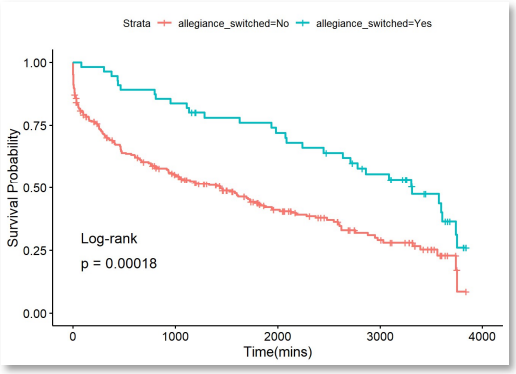
We then proceed to explore the individual survival of the different groups in the dataset.

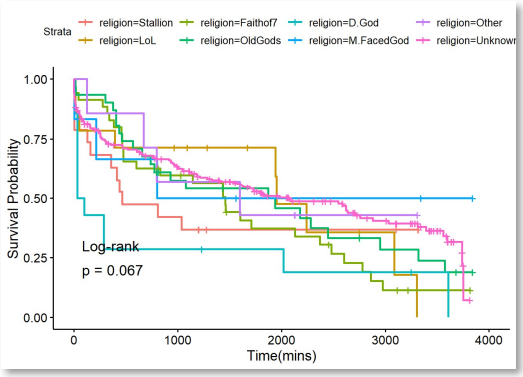
```
fitSex <- survfit(y ~ sex, data = dat)
fitOccupation <- survfit(y ~ occupation, data = dat)
fitAllegiance <- survfit(y ~ allegiance_last, data = dat)
fitSwitch <- survfit(y ~ allegiance_switched, data = dat)
fitReligion <- survfit(y ~ religion, data = dat)
fitProminence <- survfit(y ~ prominence, data = dat)
fitStatus <- survfit(y ~ social_status, data = dat)

ggsurvplot(fitSex, xlab="Time(mins)", ylab="Survival Probability", pval=TRUE, pval.method = TRUE)
ggsurvplot(fitOccupation, xlab="Time(mins)", ylab="Survival Probability", pval=TRUE, pval.method = TRUE)
ggsurvplot(fitAllegiance, xlab="Time(mins)", ylab="Survival Probability", pval=TRUE, pval.method = TRUE)
ggsurvplot(fitSwitch, xlab="Time(mins)", ylab="Survival Probability", pval=TRUE, pval.method = TRUE)
ggsurvplot(fitReligion, xlab="Time(mins)", ylab="Survival Probability", pval=TRUE, pval.method = TRUE)
ggsurvplot(fitStatus, xlab="Time(mins)", ylab="Survival Probability", pval=TRUE, pval.method = TRUE)
ggsurvplot(fitProminence, xlab="Time(mins)", ylab="Survival Probability", pval=TRUE, pval.method = TRUE)
```

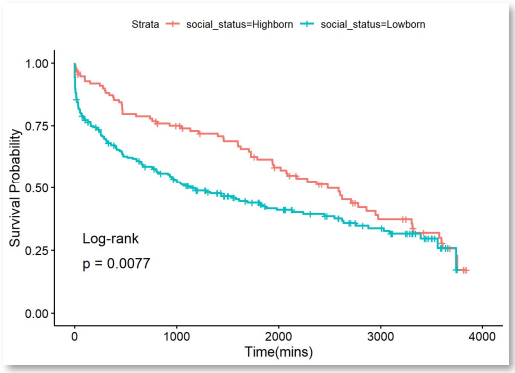

Strata sex=Male sex=Female
Survival Probability
1.00
0.75
0.50
0.25
0.00
Log-rank
p = 0.00017
0
1000
2000
3000
4000
Time(mins)

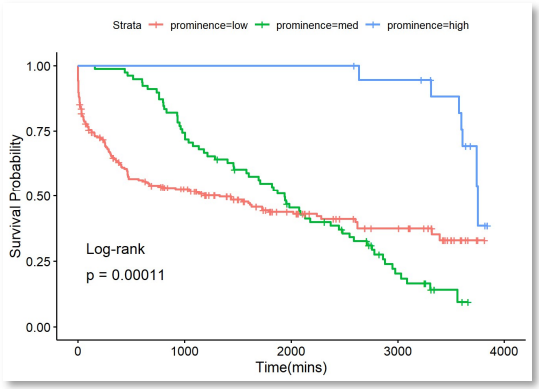

Strata occupation=Silk collar occupation=Boiled leather collar occupation=Unknown
Survival Probability
1.00
0.75
0.50
0.25
0.00
Log-rank
p = 0.028
0
1000
2000
3000
4000
Time(mins)


ice_last=Stark allegiance_last=NightWatch allegiance_last=Greyjoy allegiance_last=Frey
ice_last=Targaryen allegiance_last=Lannister allegiance_last=Bolton allegiance_last=Other
Survival Probability
1.00
0.75
0.50
0.25
0.00
Log-rank
p < 0.0001
0
1000
2000
3000
4000
Time(mins)


Strata allegiance_switched=No allegiance_switched=Yes
Survival Probability
1.00
0.75
0.50
0.25
0.00
Log-rank
p = 0.00018
0
1000
2000
3000
4000
Time(mins)


Strata religion=Stallion religion=Faithof7 religion=D.God religion=Other
religion=LoL religion=OldGods religion=M.FacedGod religion=Unknown
Survival Probability
1.00
0.75
0.50
0.25
0.00
Log-rank
p = 0.067
0
1000
2000
3000
4000
Time(mins)


Strata social_status=Highborn social_status=Lowborn
Survival Probability
1.00
0.75
0.50
0.25
0.00
Log-rank
p = 0.0077
0
1000
2000
3000
4000
Time(mins)

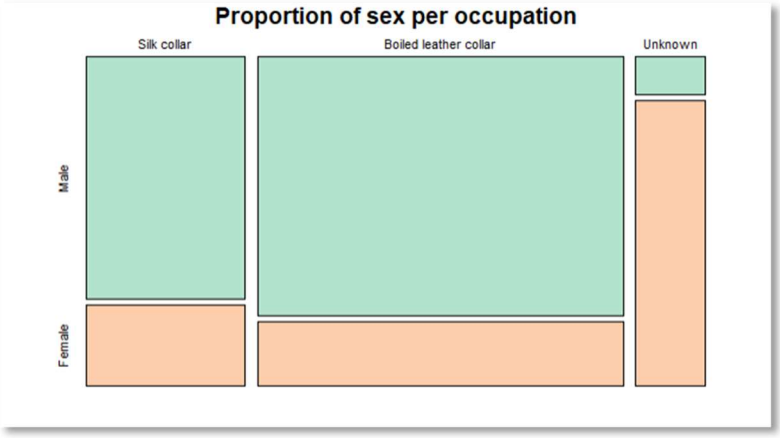

Strata prominence=low prominence=med prominence=high
Survival Probability
1.00
0.75
0.50
0.25
0.00
Log-rank
p = 0.00011
0
1000
2000
3000
4000
Time(mins)

After looking at the curves we can see that, based on the log-rank test, we can reject the no difference between groups hypothesis for almost all the covariates (log-rank test p-value is added to the survival curves plots) The only the exception is for the covariate `religion` (`p-value= 0.067`) where we can't reject the null hypothesis and therefore leave it outside any interpretation analysis.

We then want to explore stratification on the significant variables to identify and control for potential confounding factors.

4.1 OCCUPATION

Exploring the occupation curves, we see that when the occupation is **Unknown** the estimated survival is much better than for the rest (a median of 2949 minutes, compared 1711 and 1541 for the others). but when looking deeper into the group composition we see that females are overrepresented in unknown and considering that `sex` is also an extremely significant group for survival (`p-value= 0.00017`) we then suspect that sex is really the factor behind the occupation covariate.



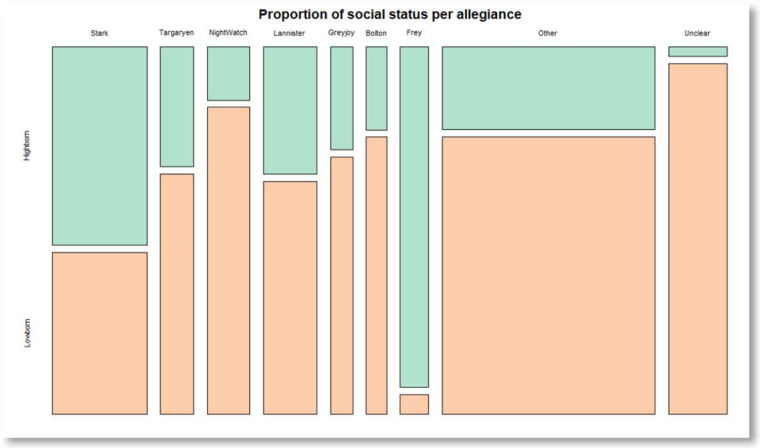
Effectively, when stratifying we see a dramatic degradation of the log-rank significance test (`p=0.4`)

```
survdiff(y ~ occupation+strata(sex), data=dat)
## Call:
## survdiff(formula = y ~ occupation + strata(sex), data = dat)
##
##                                  N Observed Expected (O-E)^2/E (O-E)^2/V
## occupation=Silk collar            96       64     59.9   0.27688   0.39254
## occupation=Boiled leather collar 221      135    134.3   0.00333   0.00967
## occupation=Unknown                42       13     17.7   1.26776   1.74837
##
##  Chisq= 1.9  on 2 degrees of freedom, p= 0.4
```

We performed the inverse analysis (`occupation` stratifying by `sex`) and we still get a significant difference between sex groups. Based on this we assume that sex is really the factor behind the difference in `occupation` groups and therefore we lose confidence that it is indeed as an interesting variable of our analysis.

4.2 SOCIAL STATUS

In principle there a significance risk difference between highborn and lowborn `social status` groups (`p-value 0.00018`), lowborn being more at risk (almost 50% more) than highborn characters. Analyzing things a little bit more carefully we find that lowborn characters are overrepresented in the allegiances with more risk (Greyjoy, Bolton, etc.)




```
coxph(y ~ allegiance_last, data = dat)
```

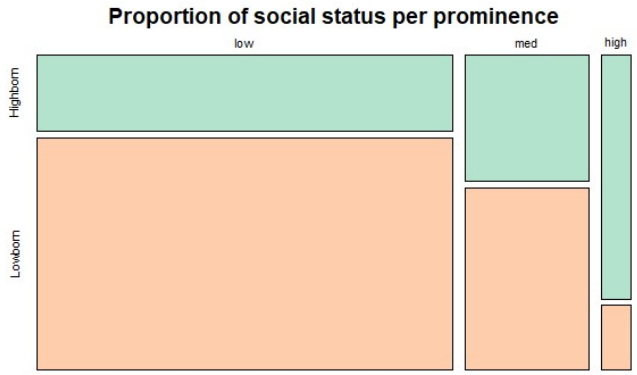
```
##                              coef exp(coef) se(coef)      z        p
## allegiance_lastTargaryen   0.2873    1.3328   0.3525  0.815 0.415119
## allegiance_lastNightWatch  0.5806    1.7871   0.3012  1.927 0.053935
## allegiance_lastLannister   0.8177    2.2653   0.2734  2.991 0.002779
## allegiance_lastGreyjoy     1.4033    4.0685   0.3656  3.838 0.000124
## allegiance_lastBolton      1.5171    4.5589   0.3471  4.371 1.24e-05
## allegiance_lastFrey       -0.4463    0.6400   0.4577 -0.975 0.329554
## allegiance_lastOther       0.5951    1.8133   0.2312  2.575 0.010033
## allegiance_lastUnclear    -0.1052    0.9002   0.3523 -0.298 0.765343
##
## Likelihood ratio test=40.42  on 8 df, p=2.677e-06
## n= 359, number of events= 212
```

```
plot(table(dat$prominence,dat$social_status),   main="Proportion of social status per allegiance", col=palette)
```

By regressing on the `prominence` covariate, we see that **high prominence** is significantly associated lower risk (`hazard ratio of 0.22`). When exploring the social status composition in terms of prominence we find that the highly prominent underrepresented in the lowborn group.

```
coxph(y ~ prominence, data = dat)
```

```
##                     coef exp(coef) se(coef)      z        p
## prominencemed  -0.04928   0.95191  0.15361 -0.321    0.748
## prominencehigh -1.49513   0.22422  0.37150 -4.025 5.71e-05
##
## Likelihood ratio test=23.69  on 2 df, p=7.178e-06
## n= 359, number of events= 212
```



With all these in consideration we the proceed stratify the log-rank test for social status by `prominence` and `allegiance`, the potential confounding factors. We see that the difference between groups is now very far from significant within the strata (`p-value 0.7`). Based on this we assume that prominence and allegiances are really the factors behind social status significance, so we lose confidence that it is indeed as an interesting variable of our analysis.

```
survdiff(y ~ social_status+strata(prominence,allegiance_last), data=dat)
##                            N Observed Expected (O-E)^2/E (O-E)^2/V
## social_status=Highborn 112       67     69.4    0.0834     0.175
## social_status=Lowborn  247      145    142.6    0.0406     0.175
##
##  Chisq= 0.2  on 1 degrees of freedom, p= 0.7
```

I've also performed other stratification combinations which all resulted in significant differences for the targeted groups.

4.3 NESTED MODELS COMPARISON

Before proceeding with the manual model creation based on the variables we found significant, we would like to perform a likelihood ratio test on the nested model to tests the assumption that variables lost confidence on due to stratification (`occupation` and `social status`) wouldn't add significance to the model:

```
fitCompNestA <- coxph(y ~ sex+allegiance_last+allegiance_switched+prominence, data = dat)
fitCompFullB <- coxph(y ~ sex+allegiance_last+allegiance_switched+prominence+occupation, data = dat)
anova(fitCompNestA, fitCompFullB)

Cox model: response is  y
 Model 1: ~ sex + allegiance_last + allegiance_switched + prominence
 Model 2: ~ sex + allegiance_last + allegiance_switched + prominence + occupation
```

	loglik	Chisq	Df	P(>|Chi|)
1	-1063.662	*NA*	*NA*	*NA*
2	-1062.110	3.103031	2	**0.2119265**

```
fitCompNestC <- coxph(y ~ sex+allegiance_last+allegiance_switched+prominence, data = dat)
fitCompFullD <- coxph(y ~ sex+allegiance_last+allegiance_switched+prominence+social_status, data = dat)
anova(fitCompNestC, fitCompFullD)
Cox model: response is  y
 Model 1: ~ sex + allegiance_last + allegiance_switched + prominence
 Model 2: ~ sex + allegiance_last + allegiance_switched + prominence + social_status
```

	loglik	Chisq	Df	P(>|Chi|)
1	-1063.662	*NA*	*NA*	*NA*
2	-1063.590	0.1436856	1	**0.7046442**

In both cases we can't reject in turn H0= "social_status coef is 0" and H0= "occupation coef is 0", therefore we assume that indeed these variables do not add any predictive information to the model.

5 DIAGNOSTICS & INTERPRETATION

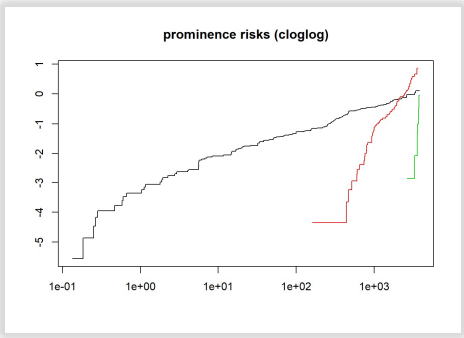
5.1 PROPORTIONALITY OF HAZARDS

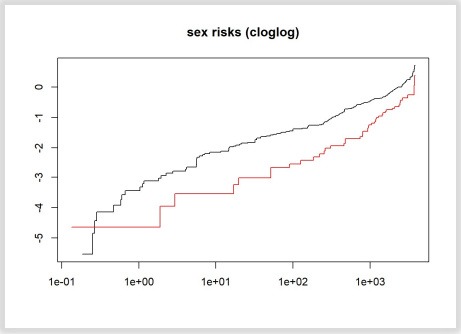
Bad news come when trying to test the proportionality of hazards assumption of the cox regression. In this case that only the `sex` covariate passes such test. Looking at the complementary log-log curves we can already see that we find some sort of parallelism only in `sex`. Diving into the proportionality of hazards assumption hypothesis test (H0 being that the risks are proportional within groups, meaning that they stay constant with time) we find that, again, we can reject this hypothesis for all covariates but `sex`.

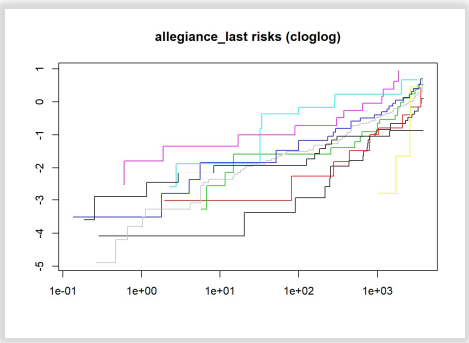
```
plot(survfit(y ~ prominence, data = dat), fun= "cloglog", col = 1:3, main="prominence risks (cloglog)")
plot(survfit(y ~ sex, data = dat), fun= "cloglog", col = 1:2,main="sex risks (cloglog)")
```

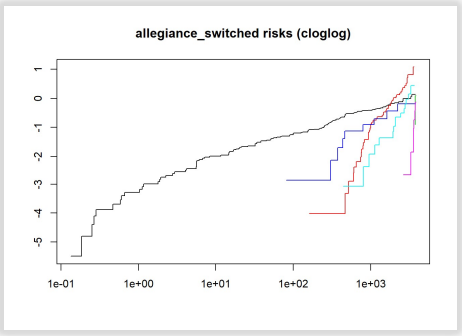
```
plot(survfit(y ~ allegiance_switched+strata(prominence), data = dat), fun= "cloglog", col = 1:6,main="allegianc
e_switched risks (cloglog)")
```

```
plot(survfit(y ~ allegiance_last, data = dat), fun= "cloglog", col = 1:9,main="allegiance_last risks (cloglog)")
```










```
cox.zph(coxph(y ~ sex+allegiance_switched+allegiance_last+prominence, data = dat))
```

```
##                      chisq df       p
## sex                    2.7  1 0.10006
## allegiance_switched   11.8  1 0.00059
## allegiance_last       25.2  8 0.00145
## prominence            66.7  2 3.3e-15
## GLOBAL                92.4 12 1.7e-14
```

Considering this we would like to try and find scenarios where we would be able to include more covariates in our model, so we explore **stratification and artificial censoring**.

5.1.1 Stratification

When exploring different stratification strategies, we get that stratifying by `prominence` brings `allegiance_switched` back into the proportionality assumption (even though not by much, `p-value: 0.0845`)

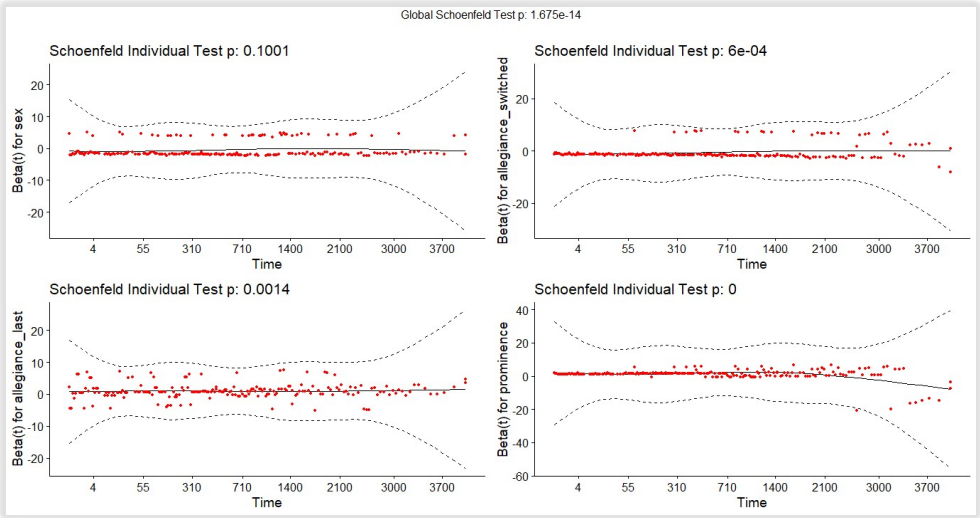
```
cox.zph(coxph(y ~ sex+allegiance_switched+allegiance_last+strata(prominence), data = dat))
```

```
##                      chisq df      p
## sex                   2.11  1 0.1464
```

```
## allegiance_switched  2.98  1 0.0845
## allegiance_last     23.53  8 0.0027
## GLOBAL              27.65 10 0.0021
```

5.1.2 Censoring

When looking at Shoenfield residuals we different residuals departure patterns that variate with time. For some covariates we see that the tail end of the experiment introduces significanct differences in the residual's departure. For prominence we see that the process starts more or less at 300 mins.



Only when we truncate our experiment to 300 mins (5 hours of show) we are able to satisfy the proportionality of hazards assumption for all the covariates:

```
new_censor_mins=300

new_censor_sec=new_censor_mins*60

dat_trunc <- within(dat, {

  dth_flag_truncated <- ifelse(exp_time_sec > new_censor_sec, 0, dth_flag)

  exp_time_sec_truncated <- ifelse(exp_time_sec > new_censor_sec,new_censor_sec, exp_time_sec)

})

cox.zph(coxph(Surv(exp_time_sec_truncated/60, dth_flag_truncated) ~ sex+prominence+allegiance_switched+allegian
ce_last, data = dat_trunc))
```

```
## Warning in fitter(X, Y, istrat, offset, init, control, weights = weights, :

## Loglik converged before variable 3,10 ; coefficient may be infinite.

##                      chisq df    p

## sex                  0.467  1 0.49

## prominence           0.862  2 0.65

## allegiance_switched  0.201  1 0.65
```

```
## allegiance_last      9.209  8 0.32
## GLOBAL              12.111 12 0.44
```

5.1.3 Conclusion

In principle we find that both stratification and artificial censoring would increase the number of explanatory covariates that we would be able to use in our model under the proportional risks assumption.

Given that the artificial censoring simulates that the experiment finishes only after 5 hs of show we would be losing a lot of information and given that we have other models that we would like to try, we lean towards keeping the original experiment censoring and use the stratified model:

```
y ~ sex+allegiance_switched+strata(prominence)
```

5.2 INTERPRETATION OF THE EFFECTS

After deciding in favor of stratifying for `prominence` to get `sex` and `allegiance_switched` to pass the proportionality of hazards we get the following model:

```
summary(fitAnalysis)
```

```
## Call:
## coxph(formula = y ~ sex + allegiance_switched + strata(prominence),
##     data = dat)
##
##   n= 359, number of events= 212
##
##                             coef exp(coef) se(coef)      z Pr(>|z|)
## sexFemale               -0.6048    0.5462   0.1707 -3.543 0.000396 ***
## allegiance_switchedYes -0.5076    0.6019   0.2142 -2.369 0.017812 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                         exp(coef) exp(-coef) lower .95 upper .95
## sexFemale                  0.5462      1.831    0.3909    0.7632
## allegiance_switchedYes    0.6019      1.661    0.3956    0.9160
##
## Concordance= 0.599  (se = 0.018 )
## Likelihood ratio test= 20.15  on 2 df,   p=4e-05
## Wald test            = 18.16  on 2 df,   p=1e-04
## Score (logrank) test = 18.65  on 2 df,   p=9e-05
```

Based on this model we can infer that our betas are indeed significant, especially for `sex`. For `sexFemale` we get beta coefficient of `-0.6048` and a hazard ratio of `0.5463`, which can be interpreted as that a character being

female in the show reduces the hazard by a little more than half. In turn risk increases by a factor of `1.831` when sex is male. (statistical significance by wald test p-value: `0.000396`)

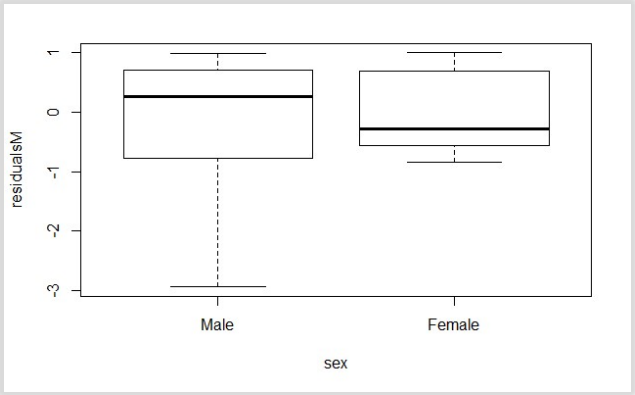
For `allegiance_switchedYes` we get a hazard ratio of `0.6019`, so we can say that the group that switched allegiance has a reduced risk (factor of 0.60) of getting killed while the ones that stay loyal have `1.661` times the risk (statistical significance by wald test p-value: `0.017812`)

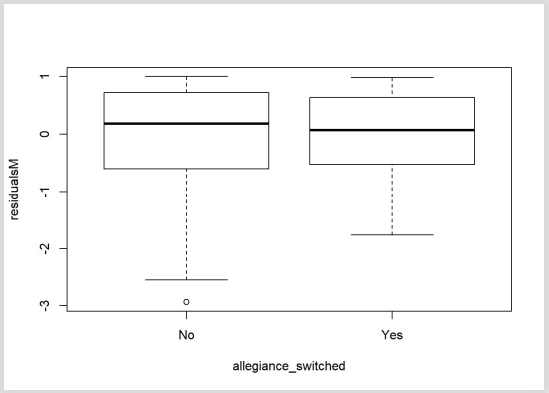
We look that we concordance index for goodness of fit we see that of `0.599` of the time the cases with the higher-risk predictor had an event before the case with the lower-risk estimated predictors.

5.2.1 Martingale residuals

We don't observe any strange distribution of the martingale residuals that would indicate non-linearity.

```
fitAnalysis<-coxph(y ~ sex+allegiance_switched+strata(prominence), data=dat)
#Martingale residuals
dat_res<-dat
dat_res$residualsM <- residuals(fitAnalysis, type = "martingale")
dat_res$residualsB <- residuals(fitAnalysis, type = "dfbetas")
with(dat_res, {
  boxplot(residualsM ~ sex)
  boxplot(residualsM ~ allegiance_switched)
})
```

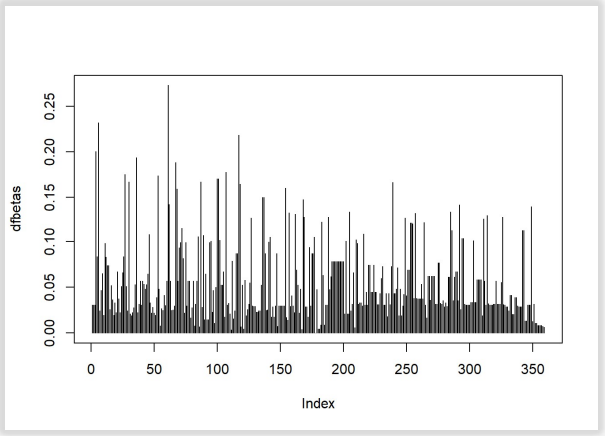




5.2.2 Outlier detection

We also don't seem necessary to remove any particular observation when looking at the beta residuals as none of them seem to be outrageously influential in the residuals that we get.

```
dat_res$residualsB <- sqrt(rowSums(dat_res$residualsB^2))
plot(dat_res$residualsB, type = 'h', ylab="dfbetas")
```



6 PREDICTION

We would now like to evaluate the prediction power model we have constructed by performing prediction against a testing dataset. We would also like to explore other models constructed by alternative variable selection, namely: Penalized regression (ElasticNet) and stepwise variable selection (using Akaike information criterion (AIC) as the performance metric.

For this exercise we will split our dataset in training and testing and we will proceed to re-train the manual model and train the alternative models for variable selection.

6.1 DATA PREPARATION

In this exercise we will be using 75% of the observations for training purposes and we will leave 25% for testing the different models.

```
#Setting seed for reproducibility
set.seed(123)
#Leave only explanatory varialbes
dat_models <- dat[,!(names(dat) %in% c("id","dth_flag","exp_time_sec"))]
#Training and testing sets
trn_size <- floor(0.75 * nrow(dat_models))
train_ind <- sample(seq_len(nrow(dat_models)), size = trn_size)
dat_trn <- dat_models[train_ind, ]
dat_test <- dat_models[-train_ind, ]
```

6.2 MANUAL MODEL

We proceed to re-train our preferred model on the training dataset to the new coefficients. When we look at the resulting model we see that both `sex` and `allegiance_switched` covariates maintain significance in the model.

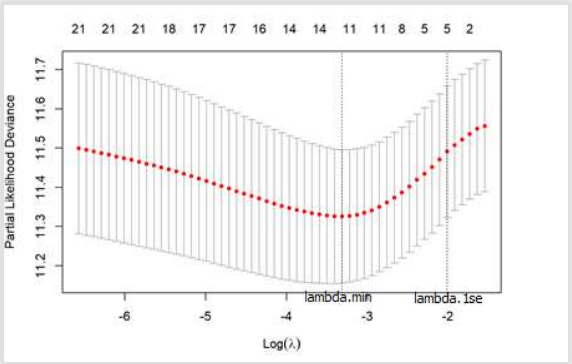
```
fitManual <- coxph(y ~ sex+allegiance_switched+strata(prominence), data=dat_trn)
summary(fitManual)
```

```
## Call:
## coxph(formula = y ~ sex + allegiance_switched + strata(prominence),
##     data = dat_trn)
##
##   n= 269, number of events= 154
##
##                             coef exp(coef) se(coef)      z Pr(>|z|)
## sexFemale               -0.5473    0.5785   0.1964 -2.787  0.00532 **
## allegiance_switchedYes -0.7250    0.4843   0.2692 -2.693  0.00708 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                        exp(coef) exp(-coef) lower .95 upper .95
## sexFemale                 0.5785      1.729    0.3937    0.8501
## allegiance_switchedYes    0.4843      2.065    0.2858    0.8209
##
## Concordance= 0.601  (se = 0.02 )
## Likelihood ratio test= 17.37  on 2 df,   p=2e-04
## Wald test            = 15.26  on 2 df,   p=5e-04
## Score (logrank) test = 15.85  on 2 df,   p=4e-04
```

```
betasManual <- coef(fitManual)
```

6.3 PENALIZED REGRESSION: ELASTICNET

We now want to build a model using elastic net. For that we start with the full model and we let the optimization algorithm to find the right penalization lambda for our equation using cross validation. In this case I would like to build two models. First one I'd consider the lambda that achieves the minimum error (`fitElasticMin`) and for the second one I'll use lambda + 1 standard error (`fitElastic1se`)



As we can see in the plot above, a different number of betas would be selected in each case (5 vs 12)

```
y <- dat_trn$y
Xmatrix <- model.matrix(y ~ ., data = dat_trn)
Xmatrix <-Xmatrix[,-c(1)] # remove intercept
fitElastic1se <- cv.glmnet(x=Xmatrix, y=y, family = "cox")
betasElasticNet1se.all <- coef(fitElastic1se, s = "lambda.1se")
betasElasticNet1se <- betasElasticNet1se.all[betasElasticNet1se.all != 0]
names(betasElasticNet1se) <- colnames(Xmatrix)[as.logical(betasElasticNet1se.all != 0)]
names(betasElasticNet1se)
## [1] "sexFemale"              "allegiance_lastBolton"
## [3] "allegiance_lastFrey"    "allegiance_switchedYes"
## [5] "prominencehigh"
fitElasticMin <- cv.glmnet(x=Xmatrix, y=y, family = "cox")
betasElasticNetMin.all <- coef(fitElasticMin, s = "lambda.min")
betasElasticNetMin <- betasElasticNetMin.all[betasElasticNetMin.all != 0]
names(betasElasticNetMin) <- colnames(Xmatrix)[as.logical(betasElasticNetMin.all != 0)]
names(betasElasticNetMin)
##  [1] "sexFemale"                "religionD.God"
##  [3] "religionM.FacedGod"       "religionUnknown"
##  [5] "occupationUnknown"        "social_statusLowborn"
##  [7] "allegiance_lastLannister" "allegiance_lastBolton"
##  [9] "allegiance_lastFrey"      "allegiance_lastOther"
## [11] "allegiance_switchedYes"   "prominencehigh"
```

6.4 STEPWISE VARIABLE SELECTION: AIC

We then compute a model using a stepwise approach for variable selection with the help of the function `step()`. In this case we will use the AIC information criterion for the selection, and we use both forward and backwards steps for the construction of the model.

At the end of the iterations we end up with a model that considers `sex` and `allegiance_switched` (similarly to our manual model) but it also adds `allegiance_last` to the mix.

```
fitAic<- step(coxph(y ~ ., data = dat_trn),direction = "both")
betasAIC <- coef(fitAic)
summary(fitAic)
```

```
## Call:
## coxph(formula = y ~ sex + allegiance_last + allegiance_switched,
##     data = dat_trn)
##
##   n= 269, number of events= 154
##
##                               coef exp(coef) se(coef)      z Pr(>|z|)
## sexFemale                 -0.45003   0.63761  0.20299 -2.217 0.026618 *
## allegiance_lastTargaryen   0.06673   1.06901  0.45064  0.148 0.882278
## allegiance_lastNightWatch  0.09172   1.09606  0.38845  0.236 0.813332
## allegiance_lastLannister   0.64959   1.91475  0.33307  1.950 0.051139 .
## allegiance_lastGreyjoy     1.27433   3.57630  0.43748  2.913 0.003581 **
## allegiance_lastBolton      1.36719   3.92432  0.38454  3.555 0.000377 ***
## allegiance_lastFrey       -0.78597   0.45568  0.57751 -1.361 0.173525
## allegiance_lastOther       0.39531   1.48485  0.28574  1.383 0.166527
## allegiance_lastUnclear     0.03700   1.03769  0.40601  0.091 0.927395
## allegiance_switchedYes    -0.87633   0.41631  0.25830 -3.393 0.000692 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##                           exp(coef) exp(-coef) lower .95 upper .95
## sexFemale                    0.6376     1.5684    0.4283    0.9491
## allegiance_lastTargaryen     1.0690     0.9354    0.4420    2.5856
## allegiance_lastNightWatch    1.0961     0.9124    0.5119    2.3468
## allegiance_lastLannister     1.9147     0.5223    0.9968    3.6781
## allegiance_lastGreyjoy       3.5763     0.2796    1.5172    8.4298
## allegiance_lastBolton        3.9243     0.2548    1.8469    8.3385
## allegiance_lastFrey          0.4557     2.1945    0.1469    1.4133
## allegiance_lastOther         1.4849     0.6735    0.8481    2.5996
## allegiance_lastUnclear       1.0377     0.9637    0.4682    2.2996
## allegiance_switchedYes       0.4163     2.4021    0.2509    0.6907
##
## Concordance= 0.69  (se = 0.02 )
## Likelihood ratio test= 54.28  on 10 df,   p=4e-08
## Wald test            = 50.79  on 10 df,   p=2e-07
## Score (logrank) test = 56.13  on 10 df,   p=2e-08
```

We get that the estimators for `sex` and `allegiance_switched` seem very significant. We also see that, in this model, there is a very significant higher estimated risk for the members of the Bolton and GreyJoy allegiances, enduring almost 4 and 3.5 times the risk of the other groups. This is concordant with what we see in the survival curves in section 4.

7 PREDICTION

With our 4 models in hand and already trained we would like to see how they perform in predicting survival in the testing set.

For that we will borrow the methodology detailed in the case study analysis introduced in the last class of the course, in which we build a single table with all the coefficients of our models, predictions and performance metrics for our models.

As we'd like to see how well the calculated coefficients predict survival in the testing set, we perform a linear transformation of our testing features matrix and our beta coefficients to get our predicted values for each of the four proposed models. With that, we would get a vector of predicted scores which in turn is a numerical continuous variable associated with the survival, interpretable only in relative terms. We then proceed to run cox regression again to test the associated of this score with survival in the testing set. We use a standard deviation rescaling for score to favor interpretation. After the regression we get our hazard ratio coefficient for the predicted score and its associated p-value.

We then proceed to get the area under the curve of the ROC curves at **t= 50 hs of show (3000 mins)** for each model, as a comparative measure of predictive power.

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


models_coefficients <- tibble(
  method = c("manual", "aic", "elasticNet1se","elasticNetMin"),
  coefficients = list(betasManual, betasAIC, betasElasticNet1se,betasElasticNetMin)
)
dat_testMatrix <- model.matrix(y ~ .-1, data=dat_test)
ytest <- dat_test$y


got_models_performance <- mutate(models_coefficients,
                            predictions = map(coefficients, ~ lincom(., dat_testMatrix)),
                            cox_obj = map(predictions, ~ coxph(ytest ~ I(. / sd(.)))),
                            cox_tab = map(cox_obj, broom::tidy)
) %>% unnest(cox_tab)


got_models_performance <- mutate(got_models_performance,
                            AUC = map_dbl(predictions, ~ survivalROC::survivalROC(Stime=ytest[, 1], status=yte
st[, 2], ., predict.time = 3000, method = "KM")$AUC)
)
```

```
got_models_performance
```

Looking at our unified `got_models_performance` tibble results we see that, even though, the model we used for interpretation (passes the proportionality of hazards check) didn't perform bad at all but the `elasticNetMin` model performed significantly better.

method	estimate	std.error	p.value	AUC
manual	0.2967092	0.1382024	0.031799875	0.6194707
aic	0.3986741	0.1348738	0.003117540	0.6220391
elasticNet1se	0.3328218	0.1427329	0.019712208	0.6258664
elasticNetMin	**0.4997371**	**0.1340464**	**0.000192936**	**0.7046483**

In `elasticNetMin` we see a better hazard ratio for the regressed score (`0.49`), the lowest standard error (`0.13`) and lowest p-values for the wald test (`0.000192936`), which indicates a very significant association of the score with survival in the testing set.

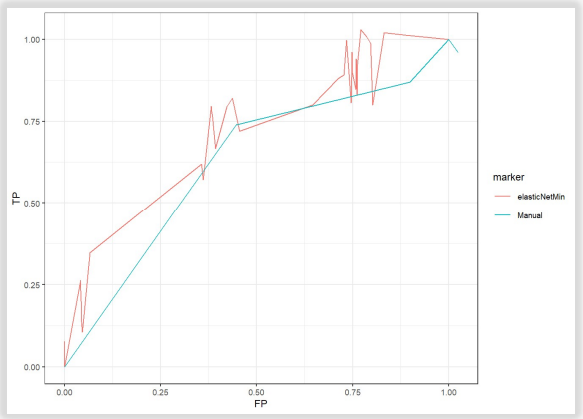
Additionally, when testing the prediction power of classifying events at time **t=50hs** we get an AUC of `0.70` for the ROC calculation, significantly outperforming all the other models.

We can see the ROC curve comparison against the manual model down below:

```
ROC_Manual<-survivalROC(Stime=ytest[, 1], status=ytest[, 2], marker= unlist(got_models_performance$predictions[
1], use.names=FALSE), predict.time = 3000, method = "KM")


ROC_Elastic_min<-survivalROC(Stime=ytest[, 1], status=ytest[, 2], marker= unlist(got_models_performance$predict
ions[4], use.names=FALSE), predict.time = 3000, method = "KM")


ROC <- list(Manual = ROC_Manual, elasticNetMin = ROC_Elastic_min)

map_dbl(ROC, "AUC")
```

```
##        Manual elasticNetMin
##     0.6194707     0.7046483
```

```
dfl <- map(ROC, ~ with(., tibble(cutoff = cut.values, FP, TP)))

for(nm in names(dfl)) {

  dfl[[ nm ]]$marker <- nm

}

ROCdat <- do.call(rbind, dfl)


ggplot(ROCdat, aes(FP, TP, color = marker)) +

  geom_line() +

  theme_bw(base_size = 9)
```



7.1 Prediction Conclusion

Even though the `elasticNetMin` model includes covariates that violate the proportionality of hazards assumption it seems to be extremely efficient in predicting survival of characters in the testing set. This model was created using an Elastic Net machine learning approach and choosing the betas associated to the lambda parameter that would give us the minimum cross validation error in the training set. The `sex` and `allegiance_switched` covariates go in the same risk direction as the ones we've identified manually, but the model has also found `religion`, `occupation`, `allegiance` and `social status` as additional significant covariates.

```
betasElasticNetMin
```

```
##                 sexFemale             religionD.God       religionM.FacedGod
##                -0.2398155                 0.7935970                -0.3583301
##           religionUnknown         occupationUnknown     social_statusLowborn
##                -0.1900544                -0.1706388                 0.1126532
## allegiance_lastLannister    allegiance_lastBolton      allegiance_lastFrey
##                 0.3203134                 0.8531770                -0.4464189
##     allegiance_lastOther   allegiance_switchedYes           prominencehigh
##                 0.0121116                -0.6061224                -0.7656406
```

When inspecting the regression on the score for model we see that for a standard deviation increase in the score regressed using `elasticNetMin`, the predicted risk increases in turn by `1.6483`.

```
got_models_performance$cox_obj[4]
```

```
## coxph(formula = ytest ~ I(./sd(.)))
##
##               coef exp(coef) se(coef)     z        p
## I(./sd(.)) 0.4997    1.6483   0.1340 3.728 0.000193
##
## Likelihood ratio test=14.35  on 1 df, p=0.0001516
## n= 90, number of events= 58
```