Capstone Report

Primary biliary cirrhosis

Simon Passek

2020-06-24

Contents

1 Introduction

Primary biliary cirrhosis(PBC), also known as primary biliary cholangitis is a slowly progressive, autoimmune disease of the intrahepatic bile system, that leads to the gradual destruction of intrahepatic bile ducts.

The typical long term effects are severe and comprise of cirrotic remodeling of the whole liver, liver inflammation, hepatic portal hypertension, liver failure, portocaval anastomoses and resulting ascites.

The typical diagnostic triade are chronic cholestatis (blockage of cholestatic secretion of the bile acid into the small intestine), frequently detectable circulating anti-mitochondrial antibodies (AMAs) (85% of patients) or other anibodies (e.g.antinuclear antibody (ANAs) and characterisitc histology at liver biopsy. It is hypothesized, that PBC often occurs in vulernable individuals, after certain enviromental exposure. The environmental triggers include toxic waste, nail polish, hair dye, various bacteria (e.g., Escherichia coli strains) and cigarette smoking. The disease prevalence is 100-fold higher in first degree relatives of a index patient, compared to average popuation risk. This suggests a stron genetic predisposition. Typically it manifests in middle-aged women (females to males 9:1) with clinical symptoms like pruritus, blood dyslipidemia, fatigue, poor disgestion with steatorrhea, nutrition deficiency syndrome, xanthomas and osteoporosis.[1] One of the

[1]Pandit et al., 2020

most widely used prognostic and treatment control scores is the Mayo Risk score.[2] Possible treatments include substitution of certain fatty acids, vitamines, and the bile Ursodeoxycholic acid. It works by reducing the concentration of the other relative toxic bile acids. The only potentially curing therapy in advanced stages is the liver transplantation, with about 70% 10-year survival after surgery. But there are always much less organs available as would be needed for transplantation. Therefore it is desirable to investigate survival outcomes of patients to identify potential prognostic biomarkers. Validated prognostic biomarkers could faciliate liver transplant allocation to those patients, for which the transplant is essenstial for survival. Another implementation of such prognostic markers would be a risk stratification, which can be used for treatment decision in the clinical setting. Paitents at high risk could be more often aided in the hospitals and included in clinical trials with new treatment options.

2 Analysis

2.1 Data import

The data for the further analysis was collected as part of a Mayo Clinic trial about PBC between 1974 and 1984. A total of 424 PBC patients met eligibility criteria for the randomized placebo controlled trial of the drug D-penicillamine, referred to Mayo Clinic during that ten-year interval. But only the first 312 cases in the dataset participated in the randomized trial. For those patients the collected data is largly complete. The data is described in (Fleming and Harrington 1991, Chapter 0.2).

Despite of held study eligibility criteria additional 112 cases contained did not participate in the clinical trial directly. From those only basic measurements and survival data was recorded. Because of that many datapoints are missing. Six of those 112 cases were lost to follow-up shortly after diagnosis and therefore excluded from the further analysis. So in summary the dataset contains are on an 312 randomized participants and additional recoreded 106 cases.

2.2 Exploratory data analysis

colnames_data	explanation
id	case number
time	number of days between registration and death, transplantion, or end of study (1986)
satus	status: 0=alive, 1=liver transplant, 2=dead
trt	drug: 1 = D_penicillamine vs. 0 = Placebo
age	age in days
sex	sex: 0=male, 1=female
ascites	presence of ascites: 0=no 1=yes
hepato	presence of hepatomegaly 0=no 1=yes
spiders	presence of ‘spiders’ 0=no 1=yes
edema	presence of edema 0=no, 0.5 = edema present without diuretics, or edema resolved by diuretics 1 = edema despite diuretic therapy
bili	serum bilirubin in mg/dl
chol	serum cholesterol in mg/dl
albumin	albumin in gm/dl
copper	urine copper in ug/day
alk.phos	alkaline phosphatase in U/liter
ast	SGOT in U/ml
trig	triglicerides in mg/dl
platelet	platelets per cubic ml / 1000

[2]Purohit et al. 2015

colnames_data	explanation
protime	prothrombin time in seconds
stage	histologic stage of disease stage 1 - 4

In summary the data consists of 17 features(biomarkers) that may be used to predict the outcome of the patient. Status = 0 describes, that at the last available follow up date (decoded as "time" from study inclusion until last follow up) the patient was alive. So in this situation one does not know wheter the patient of interest suffered the event(in this case death or transplantation) after loss of follow up, or wheter he is a long term surviver. This kind of data is called right cencored data. Status = 1 means that the patient was liver transplanted after "time" in days after study inclusion. Status = 2 decodes for death of the certain patient after "time" following after study inclusion. There are some missing values in the data set, especially for the last 106 patients, that were not included in the study:

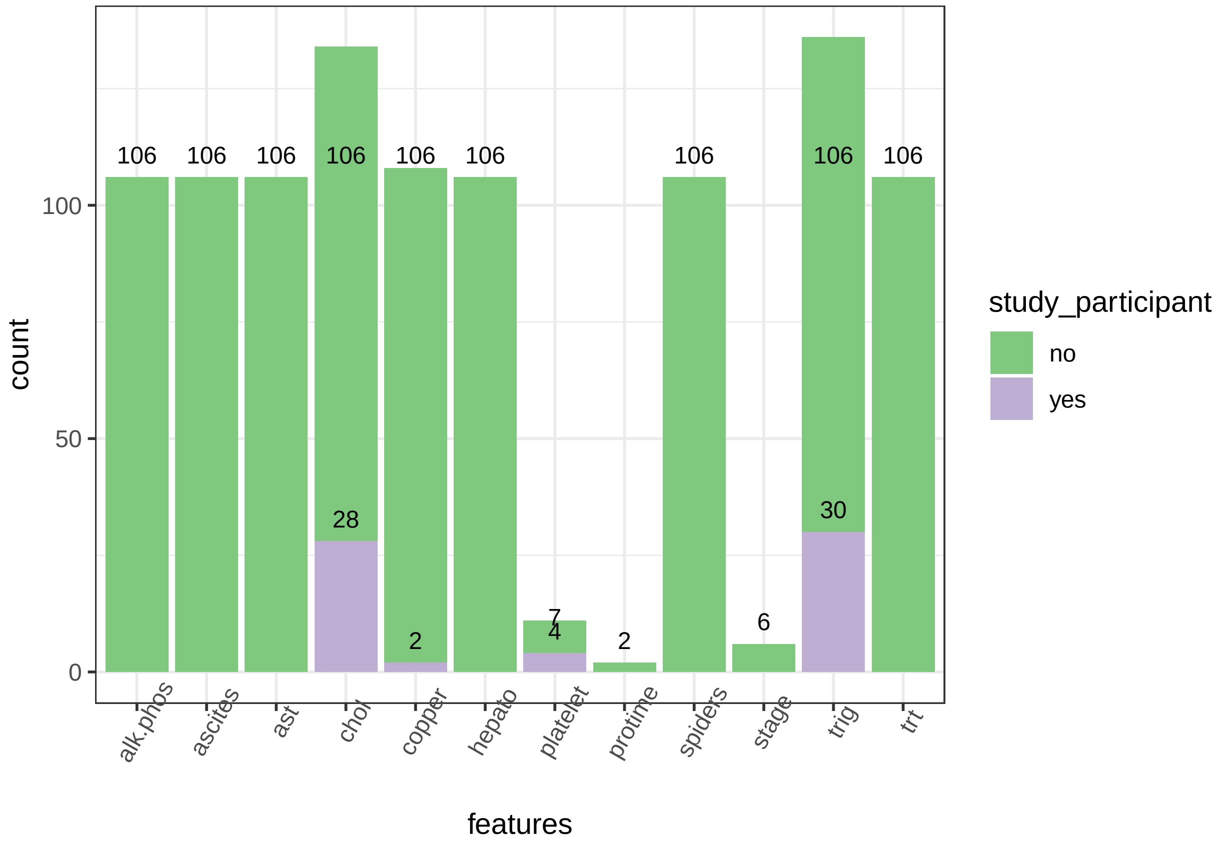


Figure 1: Number of NA values in for the 312 study patients

Only a few of the patients, that were enrolled in the study, have missing values in the features chol, chopper, platelet and trig. Here a 'NA' imputation seems justified and feasible. All non_study patients have missing data about treatment and eight other variables. Event status and time until event is avialable, but possilbly important features are missing.

```
## Warning: Removed 603 rows containing non-finite values (stat_bin).

## Warning: Removed 603 rows containing non-finite values (stat_bin).
```

Especially the frequency distributions of the enzymes alc.phos and asat, but also the blood coagulation parameter protime, the fatty blood acids parameters trig and chol and the blood cell degradation product

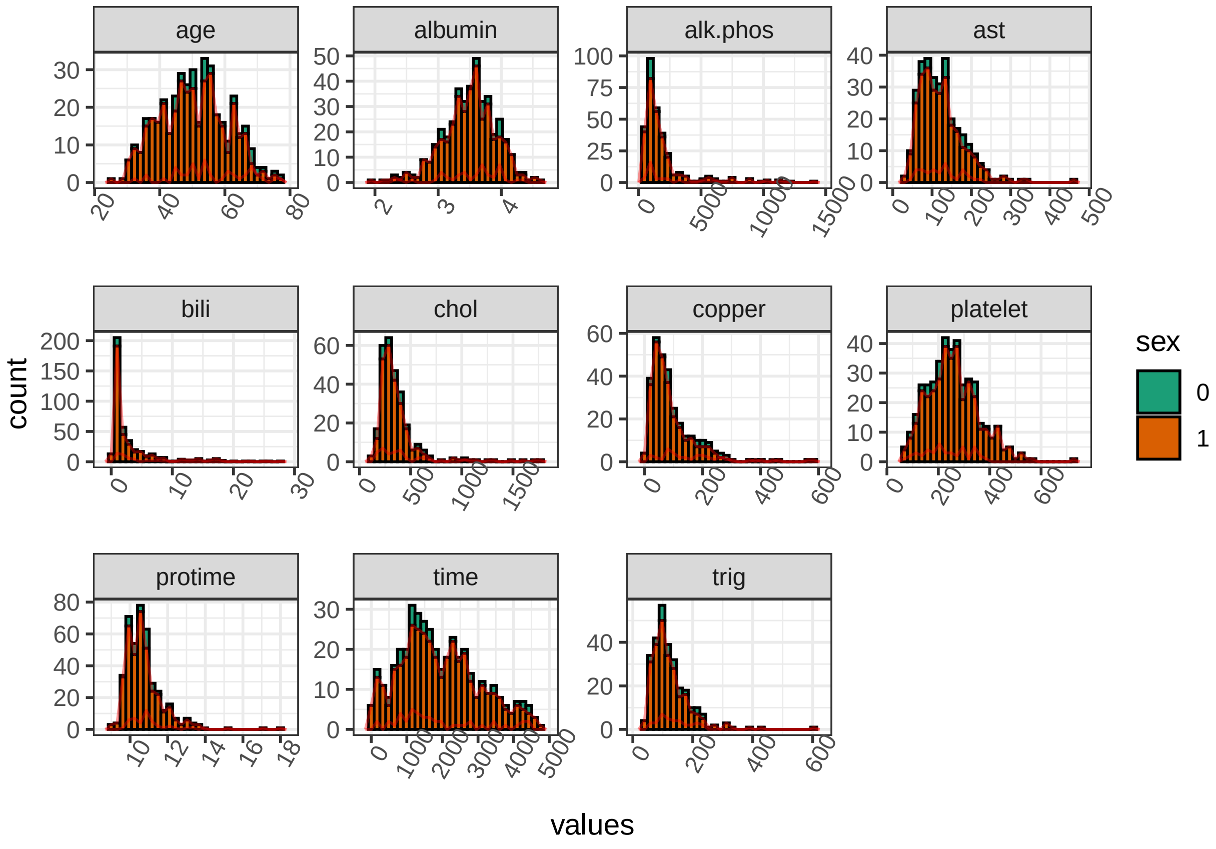


Figure 2: Histogram of continous variables

bili are skewed. This can have many possible reasons. Some of the highest or lowest values may be due to measurement error or they represent blood variables with a non normal distibution in the underlying population, as it is often the case in biological variables.

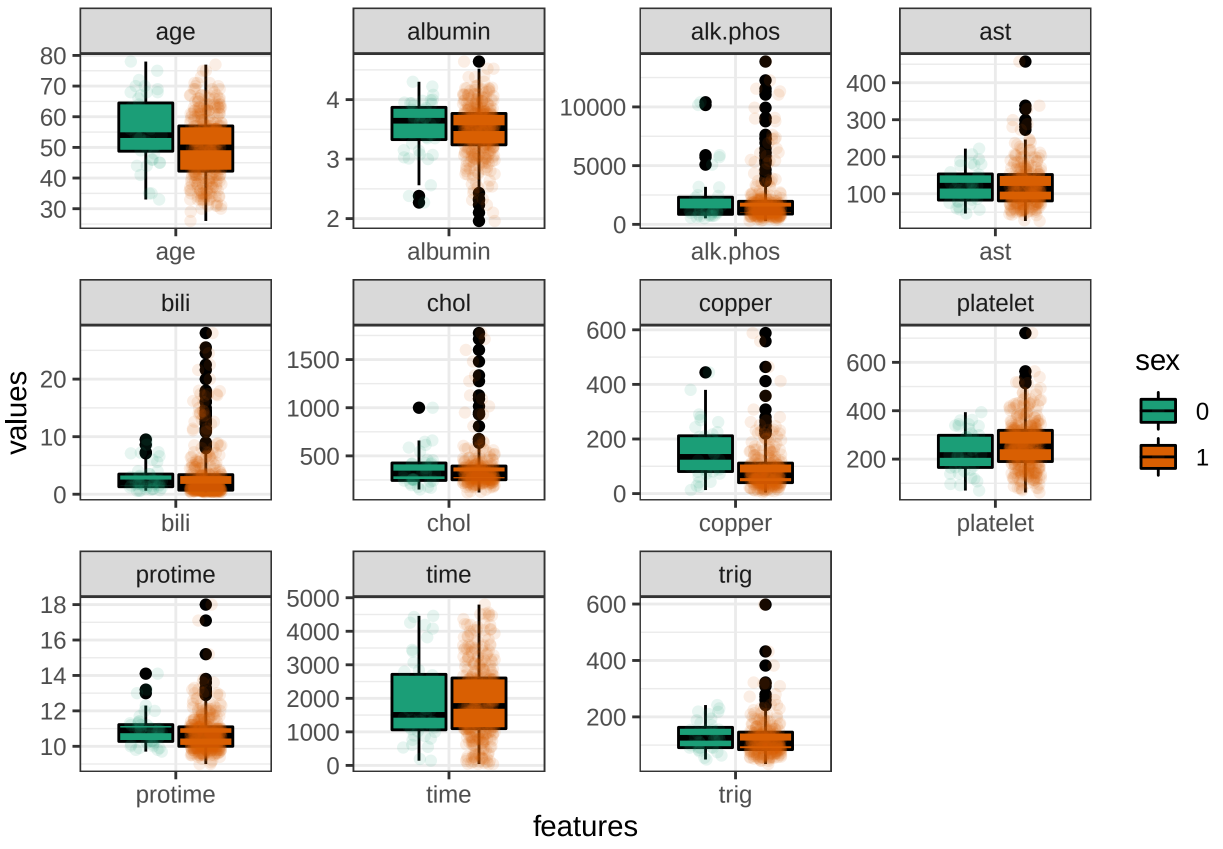


Figure 3: Boxplot of continous variables

Nevertheless transformation of those features may result in lower error rate in predictions of events later on. Especially because later used learner and prediction algrithm like the Cox Proportional Hazard model can be negavely affected by outliers.

After log2 transformation the data seems more normally distributed and outliers are partly "smoothed".

The patients in the study are classified into four histologically defined disease stages: the further the PBC has advanced, histological stages tend to increase. Most of the patients had no event (status = 0) in the available follow-up period. Few of the patients have gone through liver transplantation (status = 1). Next we will explore correlation between the quantitative variables:

There seems a positive correlation between bili and copper, ast and bili. Other stronger correlations can not be observed.

2.3 Analysis and results

The 17 possibly predictive features available were recorded at time of study inclusion. The goal of the following analysis is to identify potential biomarkers that allow risk stratification of patients into high and low risk groups. Identification of such markers may not only help to understand disease process and progression but may also allow more individual treatment decisions. A patient with a validated high risk score could be treated more intense. In the case of PBC this could mean inclusion in clinical drug trials or earlier scheduling of liver transplantation.

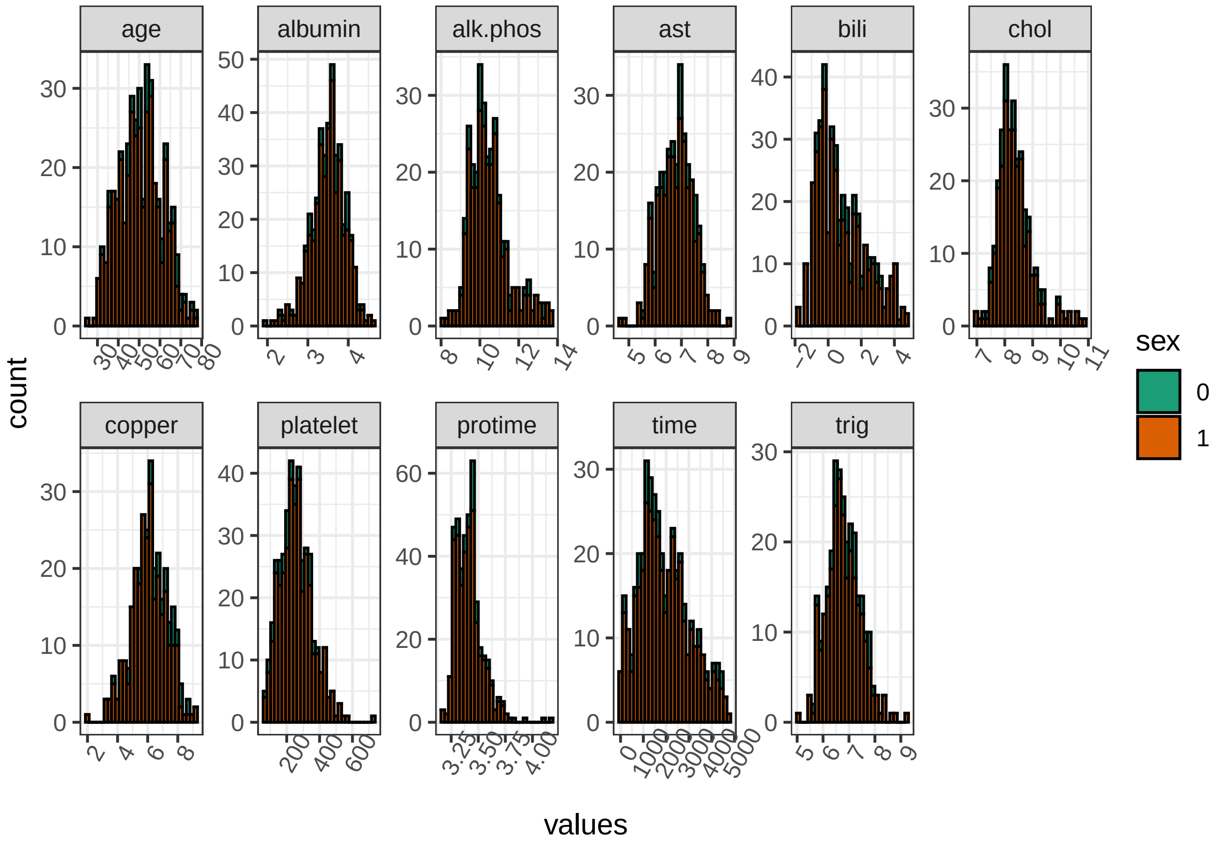


Figure 4: Boxplot of log2() transformed continous variables

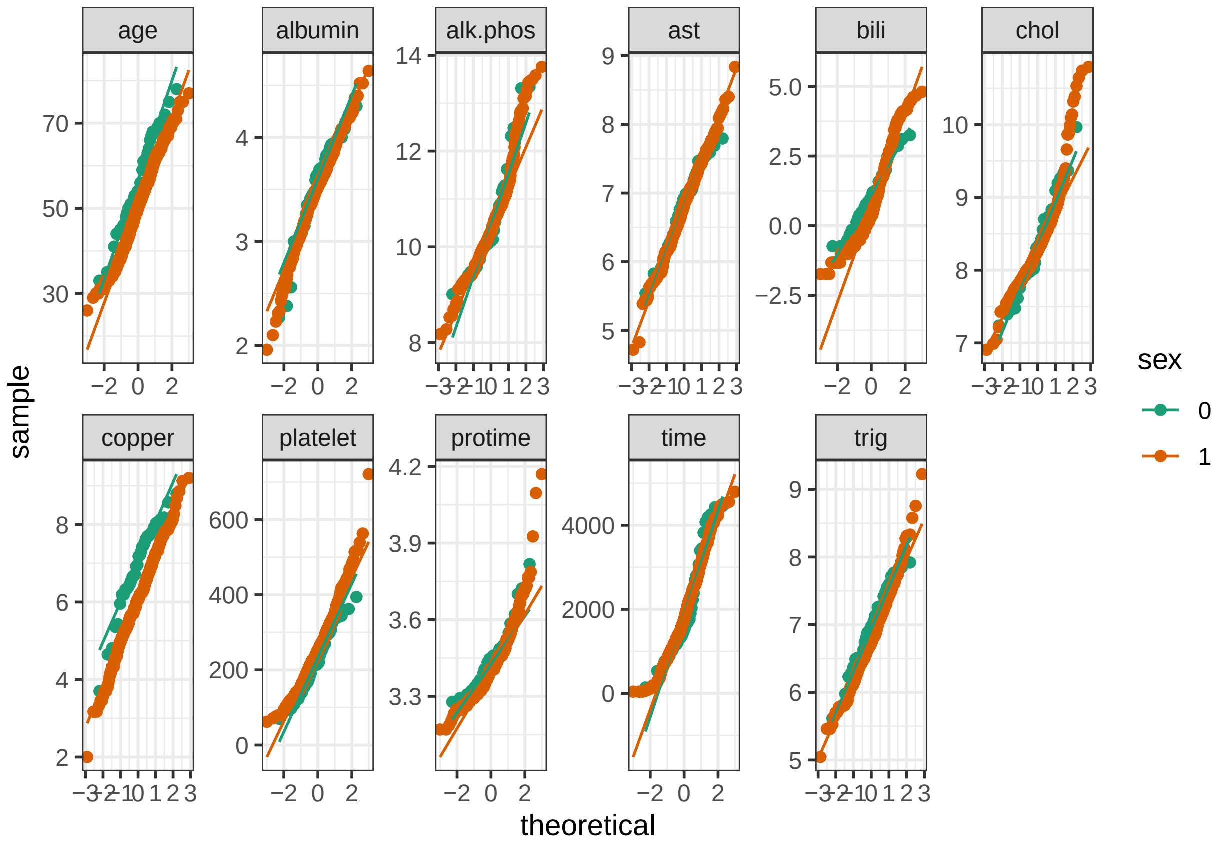


Figure 5: Quantile-quantile plot of log2() transformed features

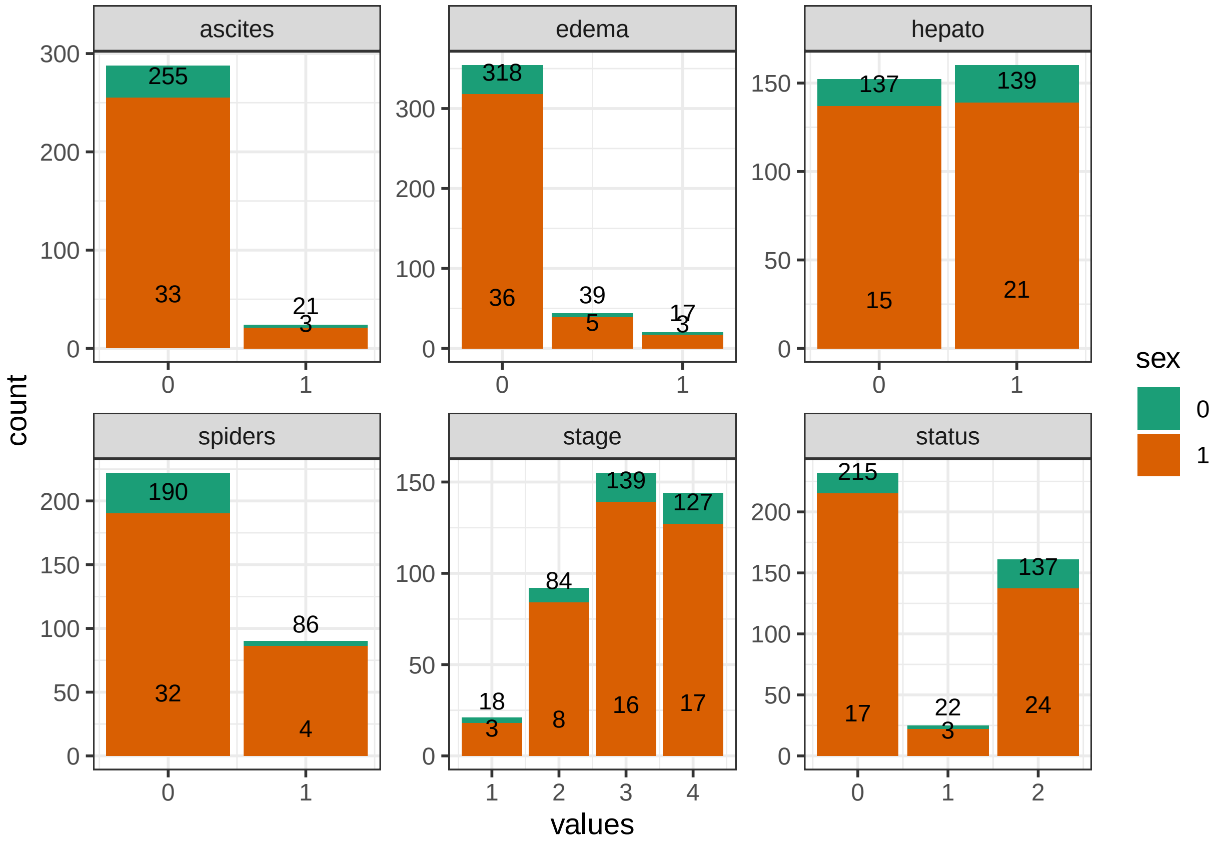


Figure 6: Barplot of categorical features

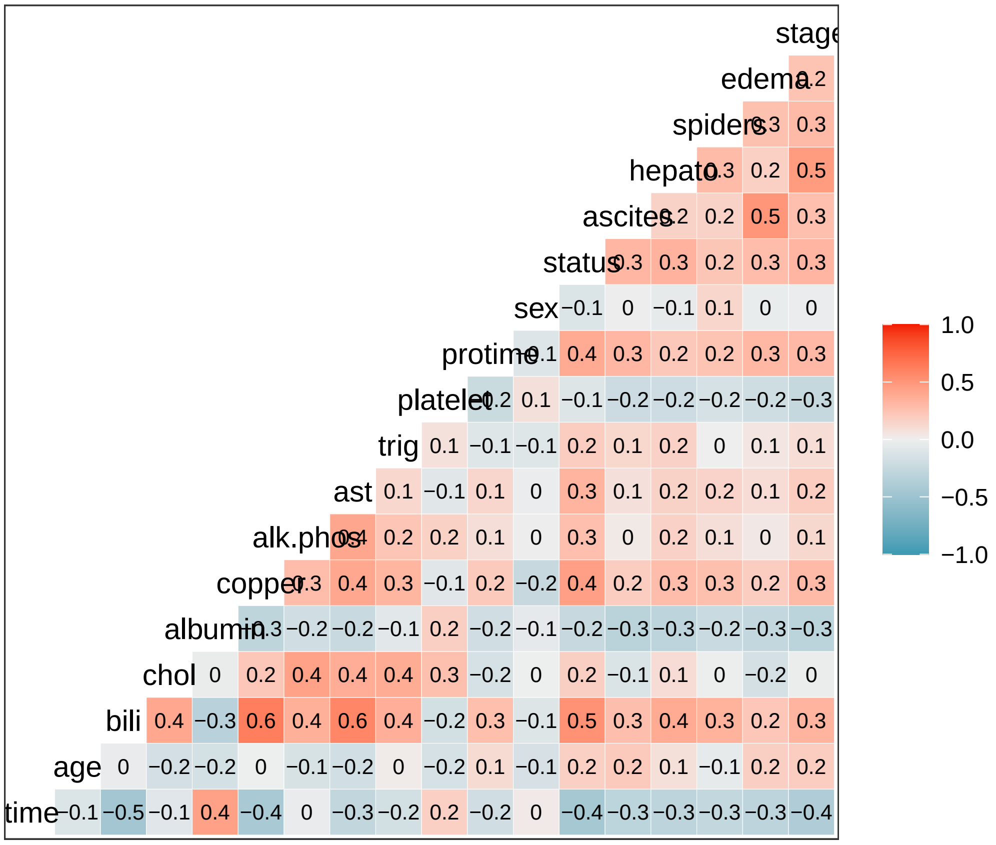


Figure 7: Spearman correlation between features

This kind of analysis is also termed survival analysis or time-to-event analysis. Survival analysis examines data on whether a specific event of interest takes place and how long it takes, until this event occurs. In this setting on can not use ordinary regression analysis. Firstly, survival data contains only positive values and therefore needs to be transformed to avoid biases. Secondly, ordinary regression analysis cannot deal with censored observations.

Censored observations are observations in which the event of interest has not occurred, yet. Survival analysis has to handle such censored, also termed right cencored data: Survival data is usually written as a pair (t_i, δ_i), where t represents the time until time of event or last follow-up, and δ is a 0/1 variable with 0= subject was censored at t and 1 = subject had an event at t. This is the most simplest setting in time-to-event analysis, because only one kind of event may or may not occur. In our case a patient with PBC is in a situation with competing risks:

1. He could get a liver transplant(status = 1).
2. He could move to another city or die in a car accident(loss of follow up, status = 0).
3. He could die before receiving a liver transplant(status = 2).

In order to simplify the following analysis, and because only 19 of the 312 randomized trial patients were actually liver transplanted, we will combine patients' status = 1 and = 0 to the status variable = 0, therefore defining them as alive at censoring time.

2.3.1 Missing value imputation and data split

For the further analysis we will first extract the 312 study participants, because there just a few variables a missing. Then by using the aregImpute function from the Hmisc package[3] is applied to impute the missing values. Another possible imputation for the missing values would be k-nearest neighbour algorithm (knn). Because of small sample sizes imputation is done on the whole dataset itself, and not on the training and testing set individually.

```
## Iteration 1 Iteration 2 Iteration 3 Iteration 4 Iteration 5 Iteration 6 Iteration 7 Iteration 8 Iter
```

```
## Iteration 1 Iteration 2 Iteration 3 Iteration 4 Iteration 5 Iteration 6 Iteration 7 Iteration 8 Iter
```

2.3.2 The Kaplan Meier estimator

2.3.2.1 Statistical background Let $Y_i(t)$, $i = 1, \dots, n$ be the indicator that subject i is at risk and under observation at time t. Let $N_i(t)$ be the step function for the ith subject, which counts the number of events for that subject up to time t.
There might me events that can happen multiple times such as rehospitalization, or something that only happens once such as liver transplantation or death. The total number of events that have occurred up to time t will be $\overline{N}(t) = \sum N_i(t)$, and the number of subjects at risk at time t will be $\overline{Y}(t) = \sum Y_i(t)$. Time-dependent covariates for a subject are the vector $X_i(t)$. It will also be useful to define $d(t)$ as the number of deaths that occur exactly at time t. The most basic describtion of time-to-event data is the Kaplan-Meier estimator.

$$S_{KM}(t) = \prod_{s<t} \frac{\overline{Y}(ts) - d(s)}{\overline{Y}(s)}$$

The Kaplan Meier estimator can also be plotted. A drop on the curve represents an event, whereas a "+" symbols a censoring. For the PBC data set the Kaplan Meier estimator can be calculated fith the `survfit()` function of the survival package [4]. One further important note: the `survfit()`function expects the event to be numeric.

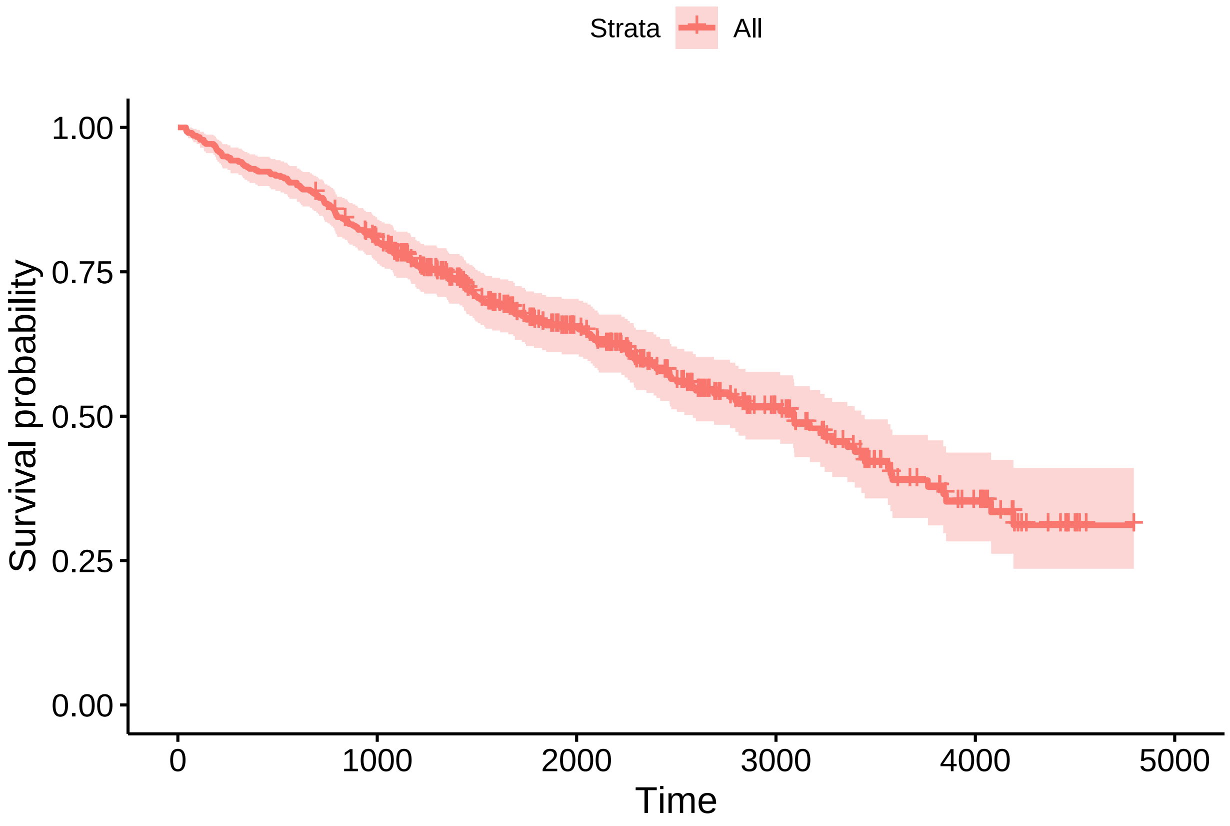


Figure 8: Survival curve of all available PBC patients

2.3.2.2 Results on the PBC data When we stratify the patients according to presence or absence of ascites we get the following Kaplan Meier plot:

```r
# plot the survival curve regarding stratified by treatment
ggsurvplot(survfit(Surv(time, status)~ascites, data = pbc_data_transformed),
           pval = TRUE, linetype = "strata", palette = c("#E7B800", "#2E9FDF"),
           risk.table = TRUE)
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```

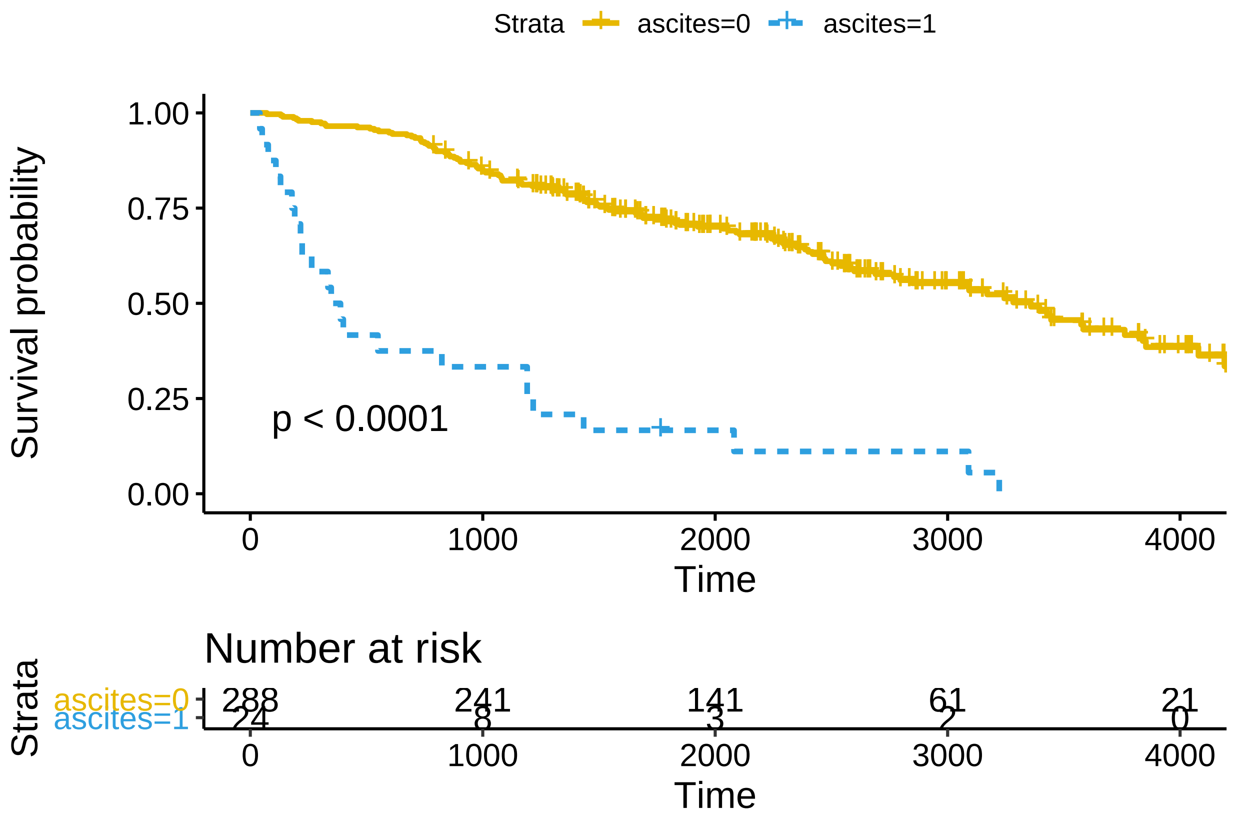


Figure 9: Survival curve stratified by presence or absence of ascites

According to this univariate analysis it seems, that the presence or absence of ascites alone has a good performance in stratifying the patients into a high and low risk group. But this is only a univariate analysis where many confounders could distort the true context. So actually, no statistically sound statment is possible, especially if you look at the patient numbers in the risk table under the Kaplan Meier plot.

2.3.3 The Cox proportional hazards model

2.3.3.1 Statistical background Other, often used mathematical models for description and prediction in multivariate survival analyses are (1)Poisson regression, (2) Cox proportional hazards model and the Aalen additive regression model. All three are closly related:

[3]Multiple Imputation using Additive Regression, Bootstrapping, and Predictive Mean Matching, https://cran.r-project.org/web/packages/Hmisc/index.html

[4]https://cran.r-project.org/web/packages/survival/index.html

(1)

$$\lambda(t|x_i) = e^{\beta_0+\beta_1 x_1+\beta_2 x_2+...}$$

(2)

$$\lambda(t|x_i) = e^{\beta_0(t)+\beta_1 x_1+\beta_2 x_2+...}$$

$$= \lambda_0(t|x_i)e^{\beta_1 x_1+\beta_2 x_2+...}$$

(3)

$$\lambda(t|x_i) = \beta_0(t) + \beta_1(t)x_1 + \beta_2(t)x_2 + ...$$

By far the most popular model is the Cox proportional hazards model, that can also written as:

$$\lambda(t|x_i) = \lambda_0(t)exp(\beta_1 x_1 + \beta_2 x_2 + ...) = \lambda_0(t)exp(x_i\beta)$$

Where $\lambda_0(t)$ is the baseline hazard function and $\beta = (\beta_1, ...\beta_K)$ is an unknown vector of regression coefficients. The baseline hazard describes how the risk of an event per time unit changes over time at the baseline levels of the covriates. The model states, that covariates are multiplicatly related to the the hazard. In a simple case a special treatment may halve the subject's hazard at any given time t, while the baseline hazard may vary.

In higher dimensional data problems, where the number of variables incease and sometimes even outgo the number of observations it can be generally important to find and subset the most relevant features of the available data. This step is also called feature selection. The purpose of such steps can be manifold:

1. The interpretability of the model and therefore for example the hypothesis generation in biological/medical questions may be eased.
2. In really high dimensional problems the model fitting and prediciton can be spneded up.
3. And maybe most importantly: by reducing the noice in the data the learner's performance and be imroved.

Different methods can be applied to feature selection. Some relevant are

- filter algorithms, they select a learner independently of the learner according so certain scores
- many machine learning methods have a built in variable importance (VIMP) measure; So selected features are important for the learner
- wrapper methods iteralively select features by themselves to optimize the performance measure

In a medical trial as the PBC is is often interesting to also gain biological understanding from the analysis and to find the variables with the most influence on the outcome. Therefore above mentioned methods can be applied, though 17 predictor variables may not be a real highdimensional problem.

2.3.3.2 Stepwise variable selection in Cox regression. Many different variable selection methods can be implemente in the context of Cox regression. One method is the stepwise variable selection using the Akaike information criteria (AIC). The AIC criterion is likelihood based and closly relates to the logarithmic scoring rule. Therefore is is considered to be adequate for identifying a prediction model. But as with any autmomated model selection procedure, results can be quite unstable. We will first use a backward stepwise variable selection implemented during the function fastbw() of the rms package.

```
# surv_form of all vailabe features

surv_form <- Surv(time, status) ~ trt +age+sex+hepato+spiders+edema+bili+
  chol+albumin+copper + alk.phos+ast+trig+platelet+protime+stage+ascites

# stepwise variable selection based on fastbw() function of the rms package
set.seed(12)
```

```
fit_cox_AIC <- pec::selectCox(surv_form, data = training_data_pbc_trial, rule = "aic")
fit_cox_AIC
```

```
## $fit
## Cox Proportional Hazards Model
##
##  rms::cph(formula = newform, data = data, surv = TRUE)
##
##                          Model Tests    Discrimination
##                                                Indexes
##  Obs       219    LR chi2    116.76    R2       0.419
##  Events     97    d.f.            2    Dxy      0.613
##  Center 0.0165    Pr(> chi2) 0.0000    g        1.420
##                   Score chi2 135.07    gr       4.135
##                   Pr(> chi2) 0.0000
##
##          Coef    S.E.   Wald Z Pr(>|Z|)
##  bili     0.9653 0.1164  8.30  <0.0001
##  albumin -0.4882 0.1076 -4.54  <0.0001
##
##
## $In
## [1] "bili"    "albumin"
##
## $call
## pec::selectCox(formula = surv_form, data = training_data_pbc_trial,
##     rule = "aic")
##
## attr(,"class")
## [1] "selectCox"
```

In the above model the coefficients age,edema and bili were selected as significant features for outcome prediction in a multivariate setting, according to Wald test with a pvalue of <0.0001.

The aregImpute() function used above, imputes the missing variables withoud centering and scaling the other continous variables, as it was done with caret::preprocess() function. In the following analysis we are going to use the aregImpute() imputed dataset.

```
# in comparison lets use areg imputed NA data---------------

# stepwise variable selection based on fastbw() function of the rms package
set.seed(12)
fit_cox_AIC_areg <- pec::selectCox(surv_form, data = training_data_pbc_trial_areg,
                                   rule = "aic")
fit_cox_AIC_areg
```

```
## $fit
## Cox Proportional Hazards Model
##
##  rms::cph(formula = newform, data = data, surv = TRUE)
##
##                          Model Tests    Discrimination
##                                                Indexes
##  Obs       219    LR chi2    127.16    R2       0.447
##  Events     97    d.f.            3    Dxy      0.633
##  Center 6.6096    Pr(> chi2) 0.0000    g        1.476
##                   Score chi2 153.17    gr       4.377
##                   Pr(> chi2) 0.0000
##
##          Coef    S.E.   Wald Z Pr(>|Z|)
##  bili     0.5789 0.0791  7.32  <0.0001
##  albumin -1.0539 0.2587 -4.07  <0.0001
##  protime  2.8723 0.8781  3.27  0.0011
##
##
## $In
## [1] "bili"    "albumin" "protime"
##
## $call
## pec::selectCox(formula = surv_form, data = training_data_pbc_trial_areg,
##     rule = "aic")
##
## attr(,"class")
## [1] "selectCox"
```

This automated feature selection algorithm may have missed some other important variables, by not combining the differend combinations and sequences of the avialable variabels. Here $2^n - 1(n = \text{number of features})$ of different feature combinations would be possible.The stepwise feature selection only 'selects' from last variable specified in the Surv() formula to the second to last and only keeps those, which significantly affect the AIC criterion.

2.3.3.3 Extension to the Cox model: Regularization

An extension of the Cox model is the addition of regularization methods such as ridge- lasso- or elastic net penalization. This allows to create a linear regression model that is penalized for having to many (possibly unimportant) variables.

In ridge regression the variables' coefficients with minor contibution to the outcome are penalized to be close to 0, but never to be 0.The penalty term to compute this is called L2-norm, which is the sum of the squared coefficients. The amount of the penalty can be fine-tuned using the tuning parameter λ : when $\lambda = 0$ the penalty term has no effect.

Lasso stands for Least Absolute Shrinkage and Selection Operator. It shrinks the variables' coefficients towards 0, using a L1-norm penalty term. L1-norom is the sum of the absolute coefficients. This can force variables with minor model contibution to be 0. This means that lasso regularization can be seen as method of feature selection.

Elastic net regression uses L1-norm and L2-norm penalization. The parameter α controls wheter more ridge or lasso regression should be used.

```
- $\alpha = 0$ -> ridge regression

- $\alpha = 1$ -> lasso regression
```

Lets fit elastic net model on the training data. The fit_enet() function is very user friendly as it autotunes λ and λ.

```
## High-Dimensional Cox Model Object
## Random seed: 5 7
## Model type: elastic-net
## Best alpha: 0.55
## Best lambda: 0.2806558
```

We can also plot the features selected by the model as a nomogram.

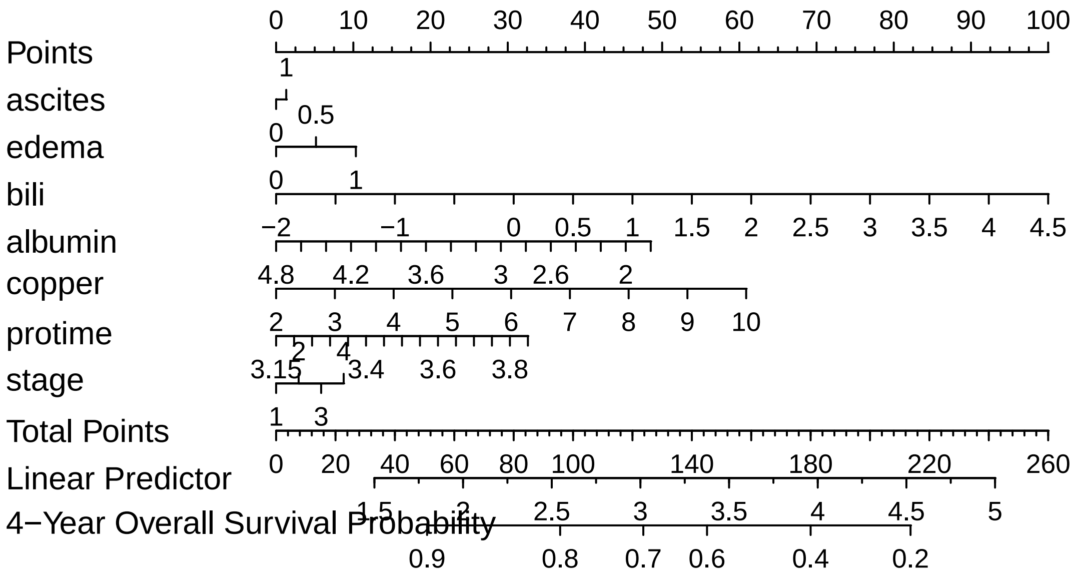


Figure 10: Nomogramm of fitted elastic net model

The elastic net regularized Cox model selected 7 variables for the final model: ascites, edema, bili, albumin, copper, protime and stage. For another statistical evaluation we will compute a unpenalized Cox model with the above selected variables.

```r
#fit multivariate unpenalized cox model on testing data
coxph(Surv(time, status)~ascites+edema+bili +albumin+protime+ stage,
      data = training_data_pbc_trial_areg)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ ascites + edema + bili +
##     albumin + protime + stage, data = training_data_pbc_trial_areg)
##
##             coef exp(coef) se(coef)      z       p
## ascites  0.29070   1.33736  0.36162  0.804  0.4215
## edema    0.54805   1.72987  0.35139  1.560  0.1188
## bili     0.54384   1.72262  0.08135  6.685 2.3e-11
## albumin -0.71795   0.48775  0.31234 -2.299  0.0215
## protime  1.75313   5.77266  1.01788  1.722  0.0850
## stage    0.29827   1.34753  0.16120  1.850  0.0643
##
## Likelihood ratio test=133.2  on 6 df, p=< 2.2e-16
## n= 219, number of events= 97
```

In this multivariate analysis the Wald test statistics and derived p-values, ascites excluded, seem to be significant on a significance level of 0.05. The variables selected by elastic net regularization seem to be significant predictors in this multivariate Cox model. This kind of model is a very popular method in medical sciences, because of its robustness and good clinical interpretability: The exponentiated coefficents 'exp(coef)' are also called *Hazard Ratios*. In the case of, for example edema, the coefficent can be interpreted as follows: A patient with pronounced edema has a 2.1 times higher chance of dying then a patient without edema. In the case of continous variables like bilirubin (bili) and because of the log2() transformation from before, a quadruplication of bilirubin value in the blood leads to an almost doubling of risk for death.

For validation purpose we can also fit a Cox model with those variables on the testing data:

```r
#fit multivariate unpenalized cox model on training data
coxph(Surv(time, status)~ascites+edema+bili +albumin+protime+ stage,
      data = testing_data_pbc_trial_areg)
```

```
## Call:
## coxph(formula = Surv(time, status) ~ ascites + edema + bili +
##     albumin + protime + stage, data = testing_data_pbc_trial_areg)
##
##            coef exp(coef) se(coef)      z        p
## ascites  0.6812    1.9763   0.5967  1.142  0.25358
## edema    0.9762    2.6543   0.7489  1.304  0.19239
## bili     0.7136    2.0412   0.1219  5.856 4.75e-09
## albumin -0.3733    0.6884   0.4077 -0.916  0.35979
## protime  0.2284    1.2566   1.3826  0.165  0.86879
## stage    0.7157    2.0455   0.2338  3.061  0.00221
##
## Likelihood ratio test=76.9  on 6 df, p=1.559e-14
## n= 93, number of events= 47
```

In the test setting only the variables stage and bili remain significant.

2.3.3.4 Validation of the elastic net regression model In order to validate the performane of the adaptive elastic net model one has to make predictions on new data and one needs a measure of performance.

In survival analysis there are many available measures. One of the most common approaches to evaluate prediction performance, especially in random Forest modeling, is Harrel's C-Index. For a good explanation of the C-Index one can refer to the article from Schmid et al.[5].

In short, a value of C near 0.5 indicates that the risk score predictions are random. Values of C close to 1 indicate, that the predicted risk scores are good at determining which patient in each of the patient pairs will have the event first. The censoring adjusted C-statistic by Uno et al. has the same interpretation as Harrel's C.

For a more continous performance parameter, that can also be depicted graphically over the time, extensions of those 'inverse-probability-of-censoring weights' based approaches like Uno's C can be made:

At each timepoint of interest Reciver operating characteristic (ROC) curves can be plotted after calculating sensitivity and specificity based on the above mentioned statistics. The area under those ROC curves (AUC) can be calculated for different time points. An AUC close to 1 means that the test sensitivity is very high, while the 1-specificity (false positive rate) is very low. One can calculate those AUCs over multiple time points and get time dependent AUC values, for each time of interest.Those can also be plotted:

```
##                365     547.5       730     912.5      1095    1277.5      1460
## Mean     0.8945274 0.8644439 0.8629691 0.8716151 0.8882853 0.8914965 0.9228326
## Min      0.8477059 0.8257960 0.8210309 0.8528441 0.8746980 0.8713538 0.9061998
## 0.25 Qt. 0.8899945 0.8614444 0.8614433 0.8700335 0.8859774 0.8912820 0.9186439
## Median   0.8967662 0.8667513 0.8675258 0.8730654 0.8897671 0.8947493 0.9256986
## 0.75 Qt. 0.9065091 0.8719428 0.8718557 0.8762231 0.8928581 0.8963247 0.9273267
## Max      0.9123825 0.8848639 0.8847423 0.8839170 0.8949847 0.8968160 0.9303129
##             1642.5      1825
## Mean     0.9204241 0.9125839
## Min      0.9025312 0.9002009
## 0.25 Qt. 0.9178957 0.9104362
## Median   0.9230169 0.9141399
## 0.75 Qt. 0.9248061 0.9159695
## Max      0.9269795 0.9188895
```

The red solid line represents the mean of the AUC, the dashed blue line represents the median of the AUC. The darker intervals in the plot show the 25th and 75th quantile of the AUC. The light intervlas show the minimum and maximum AUC values. The bootstrap based approach seems stable, as the median and mean AUC are close over all time points. For a more meaningful performance evaluation we will calculate Uno's time dependent AUCs for the test data.

```
##         91.25     182.5   273.75       365    456.25     547.5    638.75
## AUC 0.9456522 0.9456522 0.978022 0.9775281 0.9340909 0.9340909 0.9235294
##           730    821.25     912.5   1003.75      1095   1186.25    1277.5
## AUC 0.8584656 0.8632129 0.8744269 0.8947991 0.9014694 0.9014694 0.9159955
##       1368.75      1460   1551.25    1642.5   1733.75      1825
## AUC 0.8900601 0.883379 0.8917157 0.8865759 0.9035355 0.9001453
```

With a AUC of >0.9 until >450 days the model first does seem useful for risk stratification in PBC. Interesting is the loss of performance between the 500th - 1300th day of prediction.

Model calibration is a method to measure how far model predictions are from the actual survival outcomes. The calibration can be assessed by plotting the predicted survival probabilities against the actual observed survival probabilities.

This depiction shows that the elastic net model performs almost equally well on the cross validated testing dataas on the training data. Only one time point after three years is depicted in the above plots. But when doing this for multiple time points the observed trend remains the same: The model seems to overestimate the risk of actual 'low risk' patients while for actual 'high-risk' patients the risk attribution seems to be

[5]https://arxiv.org/pdf/1507.03092.pdf

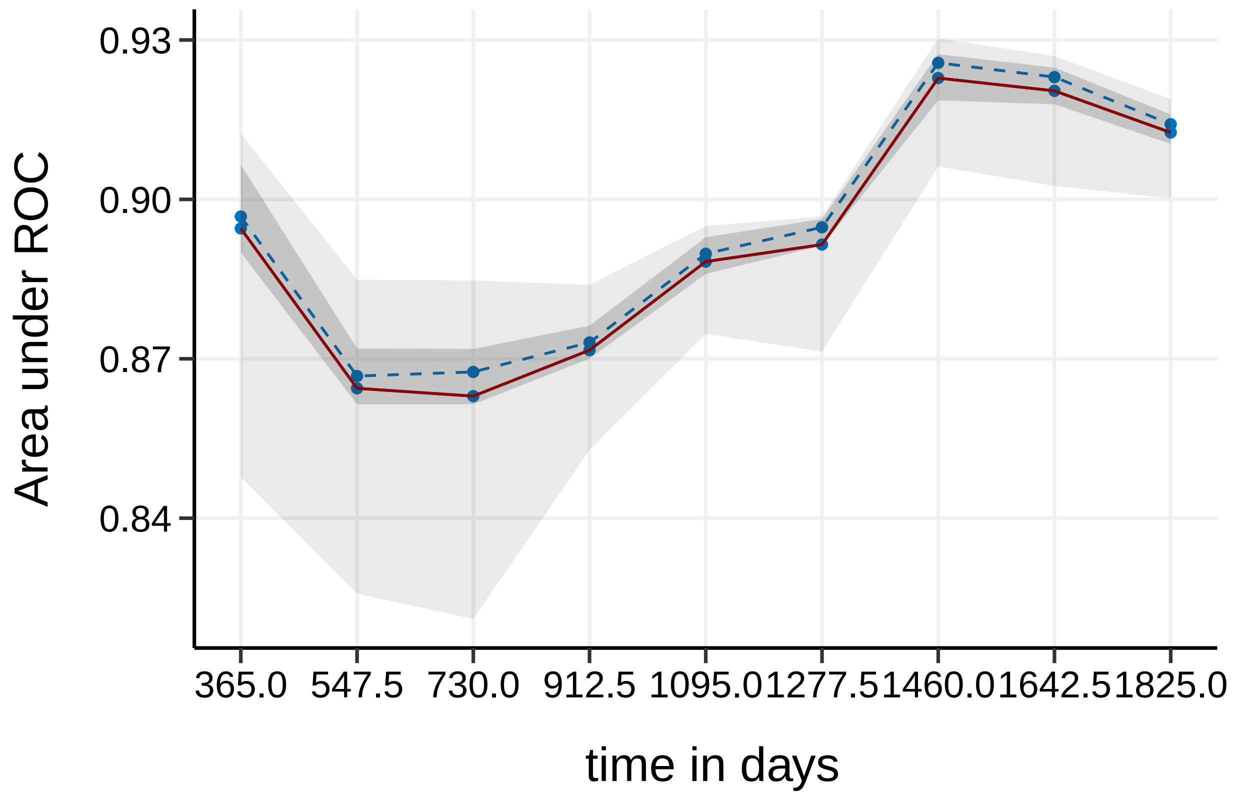


Figure 11: Uno’s time dependent AUC in training data

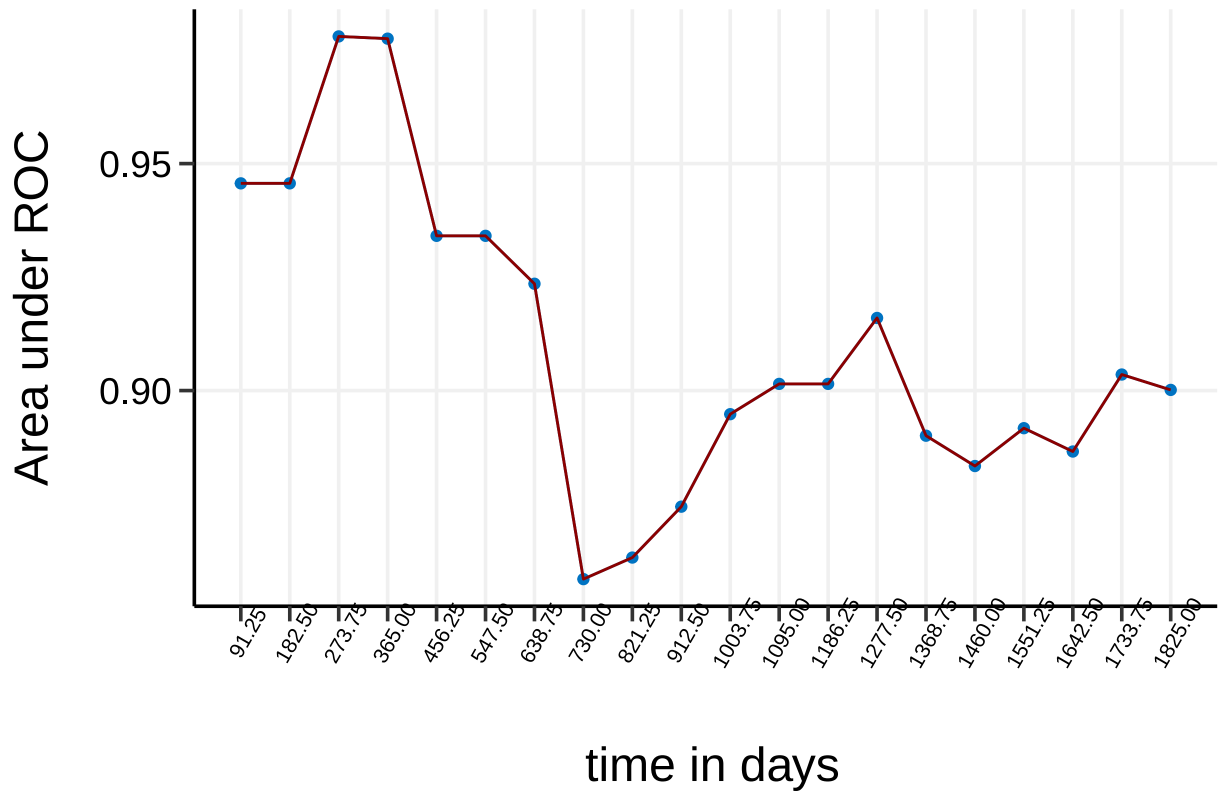


Figure 12: Uno's time dependent AUC in training data

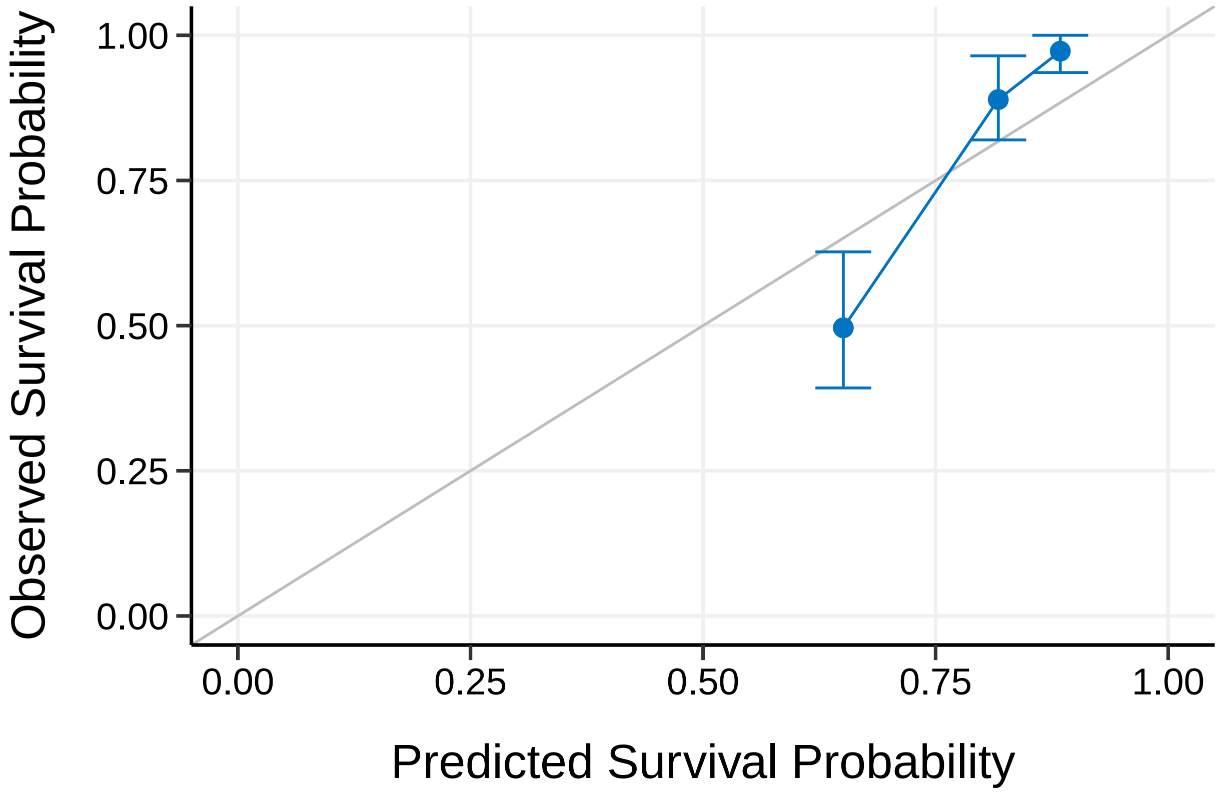


Figure 13: Internal calibration plot at 3 years

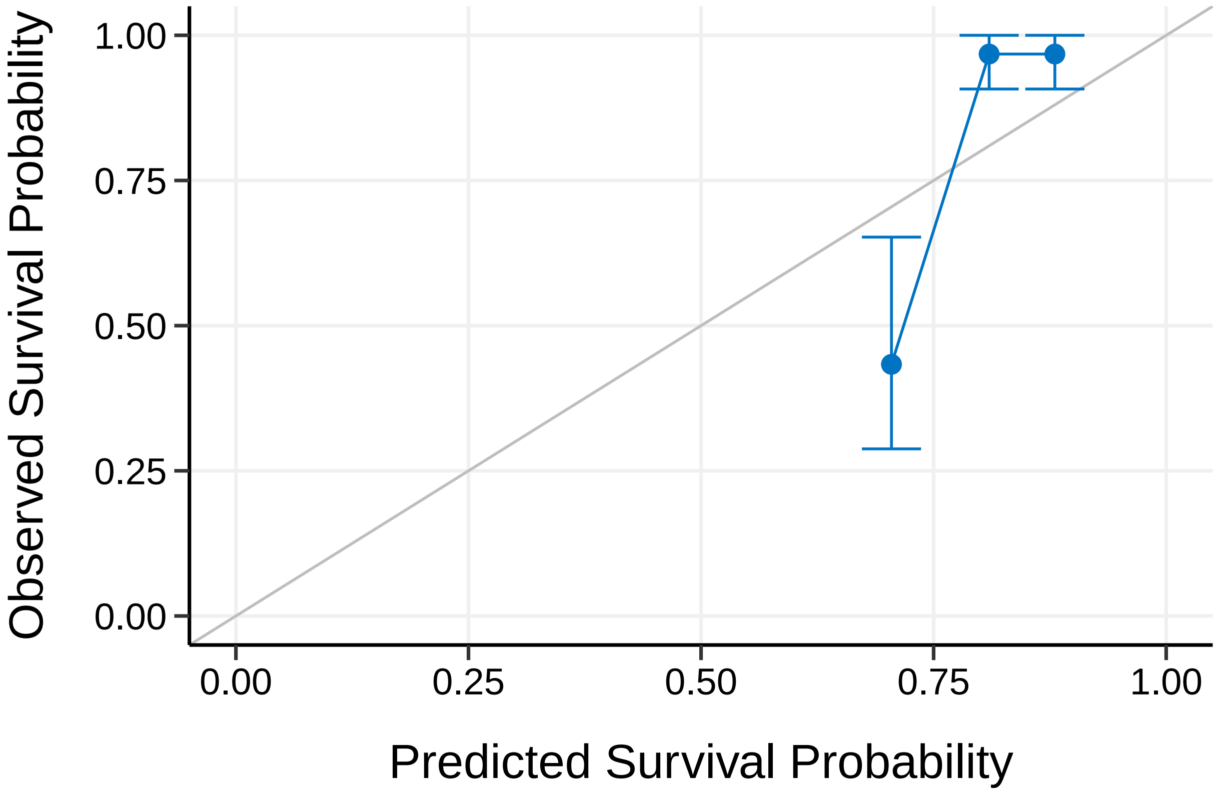


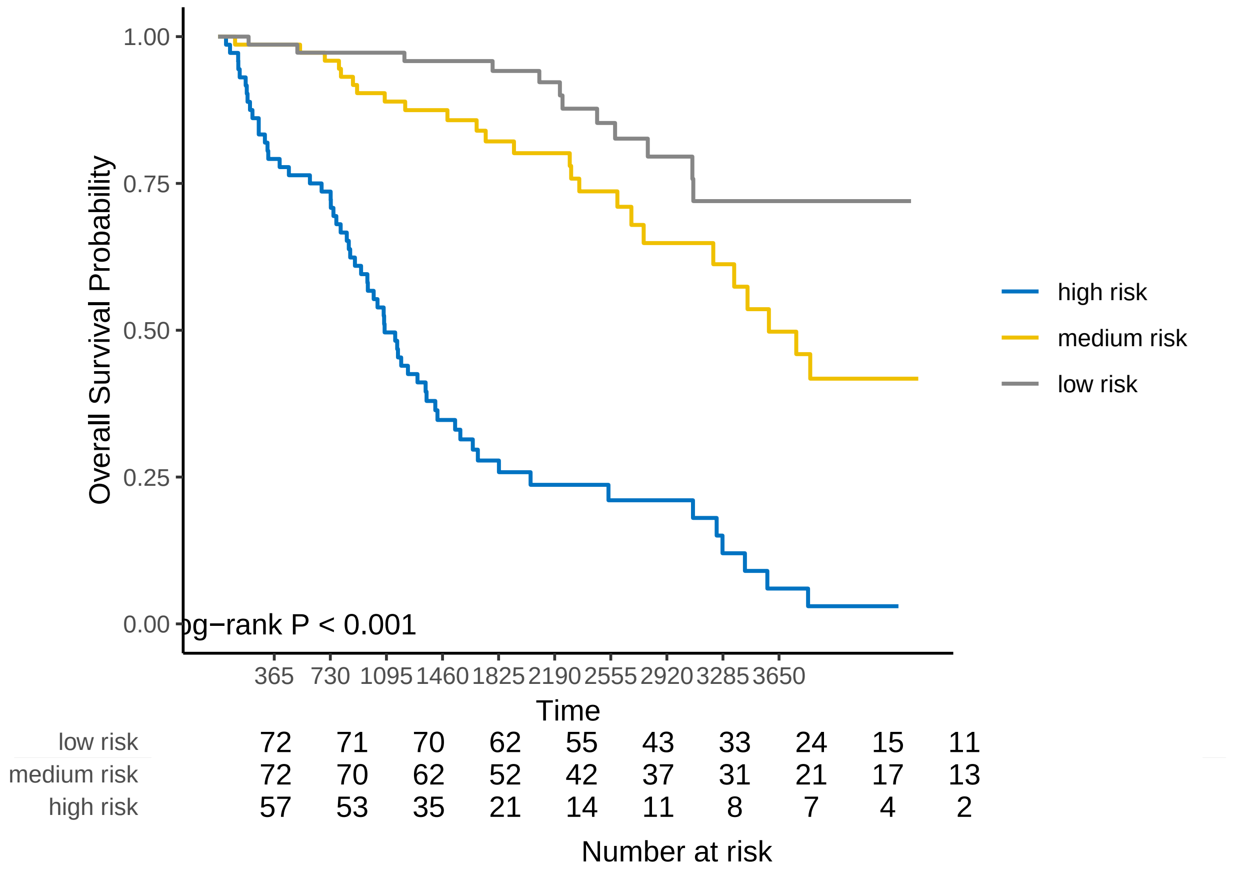
Figure 14: External calibration plot at 3 years

better. With the variable ‘ngroup’ in the calibrate() function one can set the number of risk groups and the patients are grouped into those.

2.3.3.5 Risk stratification based on the fitted elastic net model As already mentioned at the beginning of the analysis chapter, the Kaplan Meier curve is a very popular depiction of time-to-event data.

In our first approach at plotting this curve we created a survival function stratified by one univariate variable ascites: It seemed that alone the absence or presence of ascites allowed a very good risk stratification of the patients. The p-value of the Log-Rank test was <0.0001. But as depicted before, those survival curves were only univariate, and therefore no statistically sound statement can be made from this simple model. Now we have built a multivariate elastic net regularized cox model, in which 10 of the possible 17 predictor variables were shrunken to 0 by lasso’s L1-norm penalty. Ascites was selected in the final model, but it was selected as the last important variable and therefore its regression coefficient was shrunken by ridge’2 L2-norm penalty towards 0 as can be seen in the above coefficient’s nomogram.

Lets plot a Kaplan Meier curve stratified by three risk groups of the fitted elastic net model:



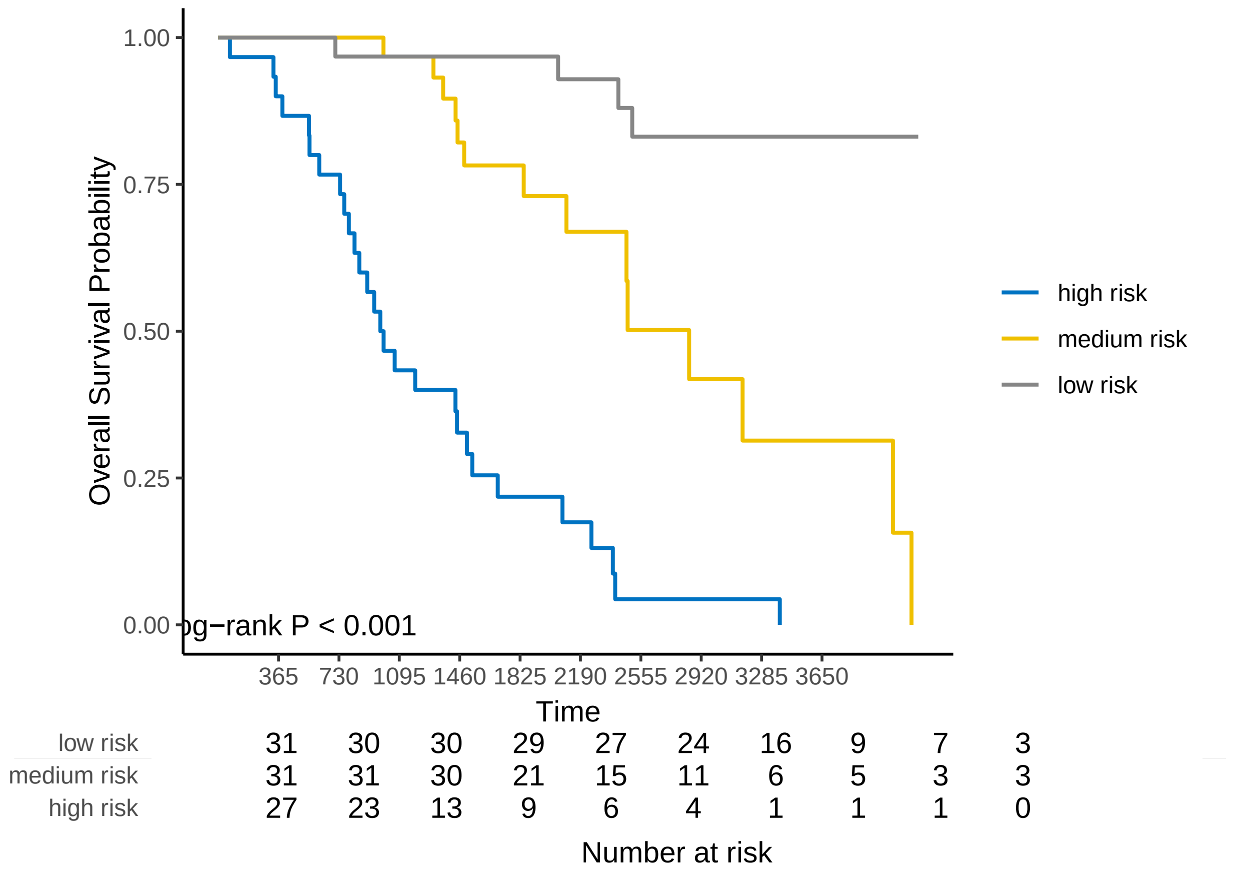
low risk	72	71	70	62	55	43	33	24	15	11
medium risk	72	70	62	52	42	37	31	21	17	13
high risk	57	53	35	21	14	11	8	7	4	2

Figure 15: Kaplan Meier curves of the training data, stratified into three risk groups

The Kaplan Meier plots clearly summarise, that the elastic net penalized cox model may has potential for usage as a prognositc scoring system in the clinical setting; for example by providing an evidence based approach for treatment decisions and treatment planning: With regard of resource allocation, for a patient in the high risk group, the liver transplantation could be planned earlier than for patients in the low risk group. Of course aided by further clinical and social variables.

The R package hdnom[6] provides a built in function for model comparison. All models that can be comared in

[6]https://github.com/nanxstats/hdnom



low risk	31	30	30	29	27	24	16	9	7	3
medium risk	31	31	30	21	15	11	6	5	3	3
high risk	27	23	13	9	6	4	1	1	1	0

Figure 16: Kaplan Meier curves of the test data, stratified into three risk groups

this environment are mathematical modifications of the described ridge, lasso and elastic net penalized Cox models. In order to find the best performing penalized Cox model for our problem they have to be compared:

CAVE: Because of a knitting failure of missing UTF8 encodings in LaTex the following plot is only visible in the html document

The adaptive elastic net may perform slightly better then the elastic net approach we fitted and evaluated above. Adaptive elastic-net models penalize the squared error loss using a combination of the L2 penalty and an adaptive weighted L1. penalty.It can be seen as a combination of the elastic-net and the adaptive lasso[7].

2.3.4 Random Survival Forests

Extensions to the random forest approach to survival analysis provide an alternative way of feature selection or building risk prediction models. The random forest approach can provide several benefits, because no parametric or semi-parametric impositions on the underlying distributions have to be made. Random forests provide an automatic way of dealing with interactions and higher-order terms in variables. The basic idea of random forest is to ensemble classification and regression trees (CART) to bootstrap sample from the training data[8]. For a final prediction the predictions of the inidvidual trees are averaged. At each node of a individual tree only a random portion of the available variables is considered for splitting. In the survival implementation of random forests the most often used maximization statistic for splitting decisions is the log-rank statistic. Ensemble predictions are given by averages over the cumulative hazard estimates in the terminal nodes of the trees, often estimated by the Nelson-Aalen estimator[9]. The most common performance measure in random survival forests is the Harrell's C-index as already mentioned above[10].

One of the most popular R implementations of random "survival" forests is provided in the randomForestSRC package[11] from Ishwaran and Kogalur.But also other R packages like party[12] implement extensions of classical random forests to survival. In the following we will work with randomForestSRC.

Let's use the package randomForestSRC to build a survival prediction model. The 'Error rate' provided by the model fit itslef is the out-of-bag (OOB) error rate, defined as 1- Harrels C-Index. This means, that a error rate close to 0 displays a good model fit, whereas values around 0.5 indicate a bad fit.

```
##                          Sample size: 219
##                     Number of deaths: 97
##                      Number of trees: 1000
##            Forest terminal node size: 15
##        Average no. of terminal nodes: 16.86
## No. of variables tried at each split: 5
##               Total no. of variables: 17
##        Resampling used to grow trees: swr
##     Resample size used to grow trees: 219
##                             Analysis: RSF
##                               Family: surv
##                       Splitting rule: logrank *random*
##        Number of random split points: 10
##                           Error rate: 18.27%
```

This random forest used all available predictor variables for training and prediction in an out-of-bag error fashion. By default the variable importance of the forest is also available.

Bilirubin, by far, seems to be the most important predictor variable for the random forest. Followed by edema, protime and copper. In comparison to the above fitted elastic net penalized cox model, which uses

[7](H.Zou et al., 2009)
[8]Breiman, L., 2001
[9]Schmidt et al., 2016
[10](Harrell et al., 1982); (Ishwaran et al. 2008)
[11]https://cran.r-project.org/web/packages/randomForestSRC/randomForestSRC.pdf
[12]https://cran.r-project.org/web/packages/party/party.pdf

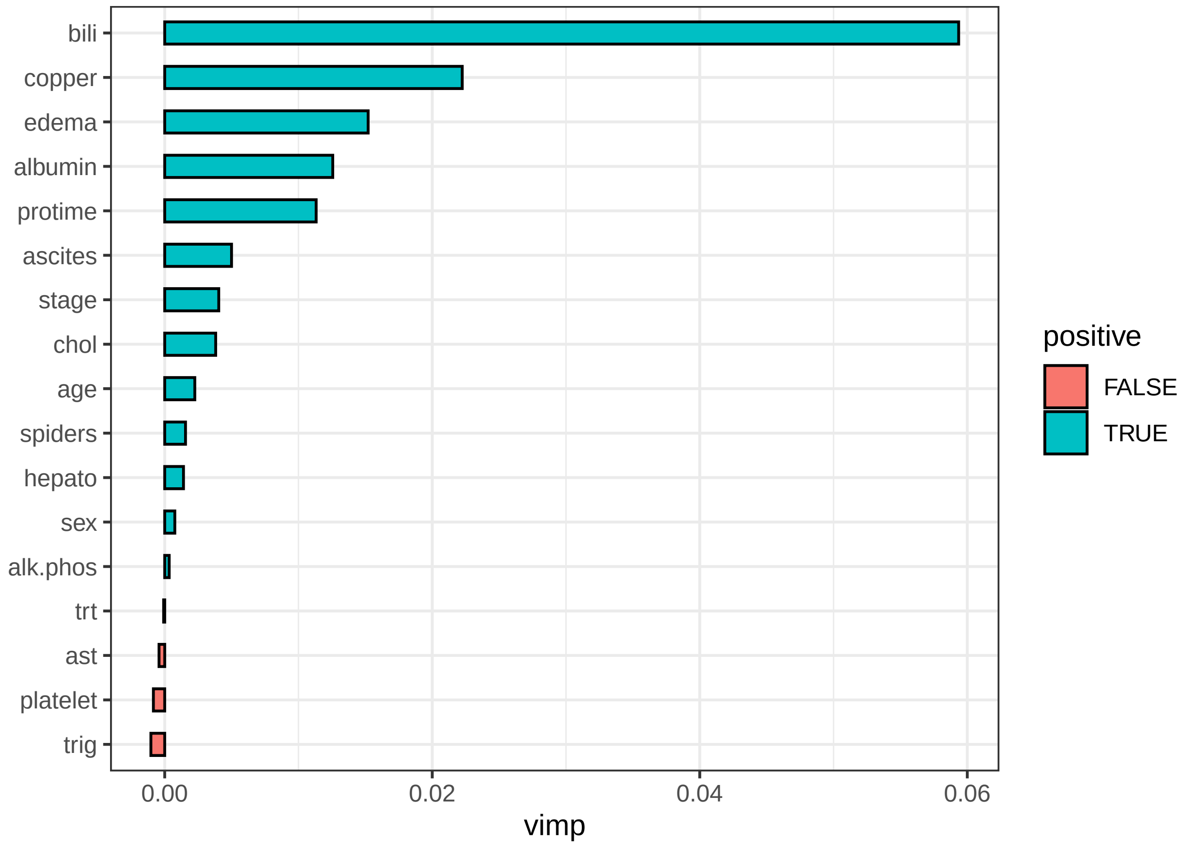


Figure 17: Variable importance of the random survival forest

the parameters bili, copper, albumin and protime as the most important variables(see nomogram). So the most important variables for both modeling approaches are analogous.

From a biological view this makes totally sense: A high bilirubin value presents a critical clinical sign, that surrogates, especially in the context of PBC, a significant, often irreversible obstruction of the bile ducts. The patients present themselves with pruritus and yellow staining of the skin and sclera. The high bilirubin values in the blood lead to many adverse events. For example damage of other organ systems and neurotoxicity. Plausibly this can lead to a higher risk of death. Edema and the parameter protime (also called protrombin time) are very good clinical markers for liver function: If the PBC has already processed far, the liver production function reduces and important blood proteins and encymes (like albumin and protime) can not be produced in an adequate amount. As a consequence the oncotic pressure in the blood system reduces and the patients develop edema.

Let's make a prediction on the test set:

```
##   Sample size of test (predict) data: 93
##        Number of deaths in test data: 47
##                 Number of grow trees: 1000
##   Average no. of grow terminal nodes: 16.86
##          Total no. of grow variables: 17
##        Resampling used to grow trees: swr
##     Resample size used to grow trees: 93
##                             Analysis: RSF
##                               Family: surv
##                  Test set error rate: 18.02%
```

Harrels C-index is 1 - Error rate which indicates a good model fit.

The package randomForestSRC also provides different ways of feature selection: - VIMP measures - minimal depth - variable hunting For further explanation refer to (randomForestSRC)[https://cran.r-project.org/web/packages/randomForestSRC/index.html] documentation. Lets use the default method minimal depth "md" so filter for relevant predictors.

```
## running forests ...
## minimal depth variable selection ...
## fitting forests to minimal depth selected variables ...
##
##
## -----------------------------------------------------------
## family             : surv
## var. selection     : Minimal Depth
## conservativeness   : medium
## x-weighting used?  : TRUE
## dimension          : 17
## sample size        : 219
## ntree              : 1000
## nsplit             : 10
## mtry               : 7
## nodesize           : 2
## refitted forest    : TRUE
## model size         : 11
## depth threshold    : 6.442
## PE (true OOB)      : 18.5889
##
##
## Top variables:
##          depth   vimp
## bili     1.401  0.020
## albumin  2.410  0.008
## protime  2.875  0.008
## copper   2.898  0.006
## chol     3.631  0.002
## platelet 3.787 -0.001
## age      4.250  0.002
## alk.phos 4.383  0.000
## edema    4.494  0.005
## ast      4.526 -0.002
## trig     4.873 -0.002
## ------------------------------------------------------------
```

```
##                          Sample size: 219
##                     Number of deaths: 97
##                      Number of trees: 1000
##            Forest terminal node size: 15
##        Average no. of terminal nodes: 12.091
## No. of variables tried at each split: 4
##               Total no. of variables: 11
##        Resampling used to grow trees: swor
##     Resample size used to grow trees: 138
##                             Analysis: RSF
##                               Family: surv
##                       Splitting rule: logrank *random*
##        Number of random split points: 10
##                           Error rate: 18.41%
```

Of the orginal 17 variables 11 were selected by minimal depth feature selection. Let's predict on the new data with the refitted tree, that is only using the eleven selected variables.

```
##   Sample size of test (predict) data: 93
##        Number of deaths in test data: 47
##                 Number of grow trees: 1000
##   Average no. of grow terminal nodes: 12.091
##          Total no. of grow variables: 11
##        Resampling used to grow trees: swor
##     Resample size used to grow trees: 59
##                             Analysis: RSF
##                               Family: surv
##                  Test set error rate: 19.93%
```

Compared to the full random forest, using all features for prediction, Harrels-C index =(1-'Test set error rate/100') decreases only a little.

2.3.5 Benchmark of different survival learners

To further compare different survival learners we will conduct a simple benchmark. The package mlr3 nad its extensions provide a machine learning environment, that includes survival learner comparison, tuning and benchmarking. Within this environment we will compare the performance of different learners on the orginal set of 17 features.

```
FALSE INFO  [00:40:37.968] Benchmark with 70 resampling iterations
FALSE INFO  [00:40:38.181] Applying learner 'surv.rpart' on task 'pbc_task' (iter 4/10)
FALSE INFO  [00:40:38.243] Applying learner 'surv.rpart' on task 'pbc_task' (iter 5/10)
FALSE INFO  [00:40:38.268] Applying learner 'surv.xgboost' on task 'pbc_task' (iter 9/10)
FALSE INFO  [00:40:38.327] Applying learner 'surv.coxph' on task 'pbc_task' (iter 10/10)
FALSE INFO  [00:40:38.445] Applying learner 'surv.glmnet' on task 'pbc_task' (iter 6/10)
FALSE INFO  [00:40:38.482] Applying learner 'surv.glmnet' on task 'pbc_task' (iter 4/10)
FALSE INFO  [00:40:38.527] Applying learner 'surv.xgboost' on task 'pbc_task' (iter 6/10)
FALSE INFO  [00:40:38.563] Applying learner 'surv.ranger' on task 'pbc_task' (iter 1/10)
FALSE INFO  [00:40:39.299] Applying learner 'surv.glmnet' on task 'pbc_task' (iter 2/10)
FALSE INFO  [00:40:39.332] Applying learner 'surv.coxph' on task 'pbc_task' (iter 9/10)
FALSE INFO  [00:40:39.383] Applying learner 'surv.coxph' on task 'pbc_task' (iter 8/10)
FALSE INFO  [00:40:39.435] Applying learner 'surv.glmnet' on task 'pbc_task' (iter 5/10)
FALSE INFO  [00:40:39.464] Applying learner 'surv.cvglmnet' on task 'pbc_task' (iter 6/10)
FALSE INFO  [00:40:39.670] Applying learner 'surv.rfsrc' on task 'pbc_task' (iter 7/10)
FALSE INFO  [00:40:39.999] Applying learner 'surv.glmnet' on task 'pbc_task' (iter 10/10)
FALSE INFO  [00:40:40.030] Applying learner 'surv.rfsrc' on task 'pbc_task' (iter 3/10)
FALSE INFO  [00:40:40.348] Applying learner 'surv.rpart' on task 'pbc_task' (iter 3/10)
FALSE INFO  [00:40:40.371] Applying learner 'surv.rpart' on task 'pbc_task' (iter 7/10)
FALSE INFO  [00:40:40.395] Applying learner 'surv.rfsrc' on task 'pbc_task' (iter 6/10)
FALSE INFO  [00:40:40.749] Applying learner 'surv.xgboost' on task 'pbc_task' (iter 1/10)
FALSE INFO  [00:40:40.772] Applying learner 'surv.rfsrc' on task 'pbc_task' (iter 2/10)
FALSE INFO  [00:40:41.118] Applying learner 'surv.glmnet' on task 'pbc_task' (iter 1/10)
FALSE INFO  [00:40:41.148] Applying learner 'surv.glmnet' on task 'pbc_task' (iter 8/10)
FALSE INFO  [00:40:41.180] Applying learner 'surv.rfsrc' on task 'pbc_task' (iter 8/10)
FALSE INFO  [00:40:41.512] Applying learner 'surv.ranger' on task 'pbc_task' (iter 10/10)
FALSE INFO  [00:40:42.277] Applying learner 'surv.rpart' on task 'pbc_task' (iter 1/10)
FALSE INFO  [00:40:42.304] Applying learner 'surv.ranger' on task 'pbc_task' (iter 3/10)
FALSE INFO  [00:40:43.031] Applying learner 'surv.glmnet' on task 'pbc_task' (iter 3/10)
FALSE INFO  [00:40:43.060] Applying learner 'surv.ranger' on task 'pbc_task' (iter 7/10)
FALSE INFO  [00:40:43.776] Applying learner 'surv.cvglmnet' on task 'pbc_task' (iter 5/10)
FALSE INFO  [00:40:43.965] Applying learner 'surv.ranger' on task 'pbc_task' (iter 4/10)
FALSE INFO  [00:40:44.695] Applying learner 'surv.rfsrc' on task 'pbc_task' (iter 10/10)
FALSE INFO  [00:40:45.064] Applying learner 'surv.cvglmnet' on task 'pbc_task' (iter 4/10)
FALSE INFO  [00:40:45.230] Applying learner 'surv.coxph' on task 'pbc_task' (iter 6/10)
FALSE INFO  [00:40:45.288] Applying learner 'surv.rpart' on task 'pbc_task' (iter 10/10)
FALSE INFO  [00:40:45.317] Applying learner 'surv.cvglmnet' on task 'pbc_task' (iter 9/10)
FALSE INFO  [00:40:45.490] Applying learner 'surv.cvglmnet' on task 'pbc_task' (iter 3/10)
FALSE INFO  [00:40:45.685] Applying learner 'surv.coxph' on task 'pbc_task' (iter 7/10)
FALSE INFO  [00:40:45.743] Applying learner 'surv.coxph' on task 'pbc_task' (iter 1/10)
FALSE INFO  [00:40:45.803] Applying learner 'surv.rfsrc' on task 'pbc_task' (iter 4/10)
FALSE INFO  [00:40:46.182] Applying learner 'surv.rpart' on task 'pbc_task' (iter 6/10)
FALSE INFO  [00:40:46.205] Applying learner 'surv.cvglmnet' on task 'pbc_task' (iter 10/10)
FALSE INFO  [00:40:46.392] Applying learner 'surv.rpart' on task 'pbc_task' (iter 2/10)
FALSE INFO  [00:40:46.424] Applying learner 'surv.ranger' on task 'pbc_task' (iter 2/10)
FALSE INFO  [00:40:47.201] Applying learner 'surv.coxph' on task 'pbc_task' (iter 4/10)
FALSE INFO  [00:40:47.249] Applying learner 'surv.rfsrc' on task 'pbc_task' (iter 5/10)
FALSE INFO  [00:40:47.577] Applying learner 'surv.coxph' on task 'pbc_task' (iter 3/10)
FALSE INFO  [00:40:47.629] Applying learner 'surv.xgboost' on task 'pbc_task' (iter 2/10)
FALSE INFO  [00:40:47.655] Applying learner 'surv.ranger' on task 'pbc_task' (iter 5/10)
FALSE INFO  [00:40:48.523] Applying learner 'surv.xgboost' on task 'pbc_task' (iter 4/10)
FALSE INFO  [00:40:48.548] Applying learner 'surv.rpart' on task 'pbc_task' (iter 9/10)
FALSE INFO  [00:40:48.571] Applying learner 'surv.cvglmnet' on task 'pbc_task' (iter 1/10)
FALSE INFO  [00:40:48.778] Applying learner 'surv.glmnet' on task 'pbc_task' (iter 7/10)
FALSE INFO  [00:40:48.812] Applying learner 'surv.xgboost' on task 'pbc_task' (iter 8/10)
FALSE INFO  [00:40:48.842] Applying learner 'surv.xgboost' on task 'pbc_task' (iter 7/10)
FALSE INFO  [00:40:48.868] Applying learner 'surv.rfsrc' on task 'pbc_task' (iter 1/10)
FALSE INFO  [00:40:49.333] Applying learner 'surv.rpart' on task 'pbc_task' (iter 8/10)
FALSE INFO  [00:40:49.366] Applying learner 'surv.cvglmnet' on task 'pbc_task' (iter 8/10)
FALSE INFO  [00:40:49.552] Applying learner 'surv.ranger' on task 'pbc_task' (iter 9/10)
FALSE INFO  [00:40:50.261] Applying learner 'surv.coxph' on task 'pbc_task' (iter 5/10)
FALSE INFO  [00:40:50.313] Applying learner 'surv.glmnet' on task 'pbc_task' (iter 9/10)
FALSE INFO  [00:40:50.345] Applying learner 'surv.xgboost' on task 'pbc_task' (iter 5/10)
FALSE INFO  [00:40:50.372] Applying learner 'surv.cvglmnet' on task 'pbc_task' (iter 7/10)
FALSE INFO  [00:40:50.574] Applying learner 'surv.coxph' on task 'pbc_task' (iter 2/10)
FALSE INFO  [00:40:50.705] Applying learner 'surv.ranger' on task 'pbc_task' (iter 8/10)
FALSE INFO  [00:40:51.450] Applying learner 'surv.xgboost' on task 'pbc_task' (iter 3/10)
FALSE INFO  [00:40:51.482] Applying learner 'surv.rfsrc' on task 'pbc_task' (iter 9/10)
FALSE INFO  [00:40:51.813] Applying learner 'surv.xgboost' on task 'pbc_task' (iter 10/10)
FALSE INFO  [00:40:51.834] Applying learner 'surv.ranger' on task 'pbc_task' (iter 6/10)
FALSE INFO  [00:40:52.558] Applying learner 'surv.cvglmnet' on task 'pbc_task' (iter 2/10)
FALSE INFO  [00:40:52.876] Finished benchmark
```

nr	resample_result	task_id	learner_id	resampling_id	iters	surv.unoAUC	surv.unoC	surv.harrellC
1		pbc_task	surv.coxph	cv	10	0.8101950	0.7677354	0.8265464
2		pbc_task	surv.rfsrc	cv	10	0.0000000	0.4543862	0.4610288
3		pbc_task	surv.xgboost	cv	10	0.7533904	0.6632069	0.7714732
4		pbc_task	surv.rpart	cv	10	0.0000000	0.6544658	0.7539809
5		pbc_task	surv.ranger	cv	10	0.0000000	0.3594790	0.3535764
6		pbc_task	surv.glmnet	cv	10	0.8216119	0.7752055	0.8335485
7		pbc_task	surv.cvglmnet	cv	10	0.8212706	0.7827490	0.8268659

Cave: In the surv.rfsrc model the C-index is automatically reported as 1-(C-index). Surprisingly the simple Cox model seems to perform as the second best compared to the other leaners.

Only the surv.glmnet leaner outperforms the simple coxph model.The surv.glmnet a learner from the glmnet package, is actually just a Cox model with elastic net regularization like the one we trained and predicted with above.

Of course the other learners are just preliminary assessed with the little benchmark above. For a further, more accurate analysis one could tune every learner and redo the benchmark. Regardless, the elastic net regularized Cox regression, but although the Cox model without regularization seems to perform well, with the default tuning parameters clearly better than the other, more complex models.

3 Summary and Conclusion

In the conducted analysis we tried to identify prognostic biomarkers which could help in risk stratification of PBC-patients. Therefore we applied different machine learning algorithms such as random forests or generalized linear models like Cox regression. The best results in risk prediction could be achieved by an elastic net penalized Cox regression model.

Interesingly the penalized Cox model outperformed the random survival forest by far. This could be due to the low dimensionality of the data, consisting of only seventeen possible predictor variables. This is not a setting for which random forests were built, since random forests were developed for multi-dimensional datasets, where the number of features can easily exceed the number of observations.

However, both approaches(random forests and Cox modelling) helped in identifying prognostically relevant blood biomarkers. In both settings the variables bilirubin, edema, prothrombin and urinary copper levels seemed to have the greatest impact on prediction performance. Those four markers seem biologically feasible in terms of disease pathophysiology.

Other markers as hepatic transferases, triglycerides and histological state, for which a patient even has to undergo biopsy, did not seem to have any prognostic impact.

This whole analysis is by far only exploratory: The analyzed data was collected from 1974 to 1984 and only containes 312 tidy annotated patients. Further we had to imput missing values.

In order to not further downsize the dataset we labeled patients with a liver transplantation as survivors. Technically this leads to bias and confounding when building models. But in order to analyze this dataset without this simplification one would have to handle it as a competing-risk time_to_event analysis. But only 19 of the 312 randmized patients were liver transplanted, a far too low number in regard to confounders. The fitted models are only some of the available models that can be used to analyze time to event data. Different methods, for example gradient boosting or other random survival forest approaches could result in better model fits in this setting.

The whole analysis is descriptive and retrospective, so prone to bias and confounding. To further validate risk scoring models in this setting one could assemble other retrospective datasets to gain a bigger analysis pool. If the variables selected as significant in the present analysis could be confirmed in this setting, one could think of conducting a prospective clinical trial, in which patients get stratified into risk groups at the day of enrollment. This is probably the only way of excluding many confounding variables one can not grasp in a retrospective setting.