University of California, Santa Barbara

Department of Probability and Statistics

PSTAT 175 Final Project

Heart Failure Survival Analysis

Report by Group A:

Gracie Shi, Jianing (Julia) Chen, Sifeng Li

Professor: Drew Carter

December 20th, 2020

Abstract

Heart failure is defined as inability of the heart to pump enough blood to sustain normal bodily functions. It is well-established that numerous researches focus on investigating heart failure problems. This study aims to analyze the survival rate of heart failure patients and determine important variables on the statistical model of identifying heart failure problems in practical settings.

To test if age, ejection fraction, high blood pressure and serum creatinine are important variables on deciding heart failure patients' survival rate, we draw Kaplan-Meier plots, then use Cox Proportional Hazard model and Analysis of Variance (ANOVA) Table to determine statistically significant parameters. Later, we perform forward selection to identify the optimal model with the smallest AIC. Furthermore, we use Random Forests and K-Nearest Neighbors to help improve proposed models by calculating the training accuracy.

We have found that the best model to use to predict heart failure patients' survival rates is *Surv ~ age+ejection fraction+serum creatinine+high blood pressure*. On this basis, information on age, ejection fraction, serum creatinine, and high blood pressure should be taken into consideration when a physician is trying to know whether the heart failure patient can survive.

Keywords: Heart Failure, Kaplan-Meier Plots, Cox Proportional Hazard Model, ANOVA Table, Forward Selection, AIC, Machine Learning, Age, Ejection Fraction, High Blood Pressure, Serum Creatinine

Data Description

This dataset is about the survival time of heart patients with various electronic medical records of symptoms, body features, and clinical laboratory test results. The dataset consists of 105 women and 194 men with ages ranging from 40 to 96 years old.

There are a total of 13 features including categorical features and numeric features without missing data:

1. anaemia (0 for no, 1 for yes): decrease of red blood cells or hemoglobin
2. high blood pressure (0 for no, 1 for yes): if the patient has hypertension
3. diabetes (0 for no, 1 for yes): if the patient has diabetes
4. sex (0 for women, 1 for men)
5. and smoking (0 for no, 1 for yes): if the patient smokes or not
6. age: age of the patient (years)
7. creatinine phosphokinase (CPK): level of the CPK enzyme in the blood (mcg/L)

8. ejection fraction(EF): percentage of blood leaving the heart at each contraction
9. platelets: platelets in the blood (kiloplatelets/mL)
10. serum creatinine (SC): level of serum creatinine in the blood (mg/dL)
11. serum sodium (SS): level of serum sodium in the blood (mEq/L)
12. time: follow-up period (days)
13. death event (0 for censored, 1 for death)

The dataset was obtained from the UCI Machine Learning Repository.

Literature Review

Before we did this project, we reviewed the case study by Ahmad and his colleagues (2017). Their work was done in the traditional biostatistics way and concluded that age, high blood pressure, ejection fraction and anemia were found as significant factors. We are more likely to jump out of this by using several data mining techniques to predict survival of the patients.

Chicco and Jurman (2020) use the case study by Ahmad et al. as a reference to build the classification. However, the limitation of their work is that they do not apply any survival analysis methodologies, instead, they use all clinic features at first time, then use two top features: serum creatinine and ejection fraction (without follow-up time and with follow-up time). Their results show that two features alone without follow-up time is the most accurate one. But we are concerned about why they would include the follow-up time when doing machine learning algorithms without clear explanations.

Research Questions & Goal

Our research question is to find the variables that are statistically significant in leading to a heart failure medical record for the patient. Besides, our goal is to use the heart failure of a medical record to build and summarise a survival predictive model which is suitable for both theoretical and practical settings.

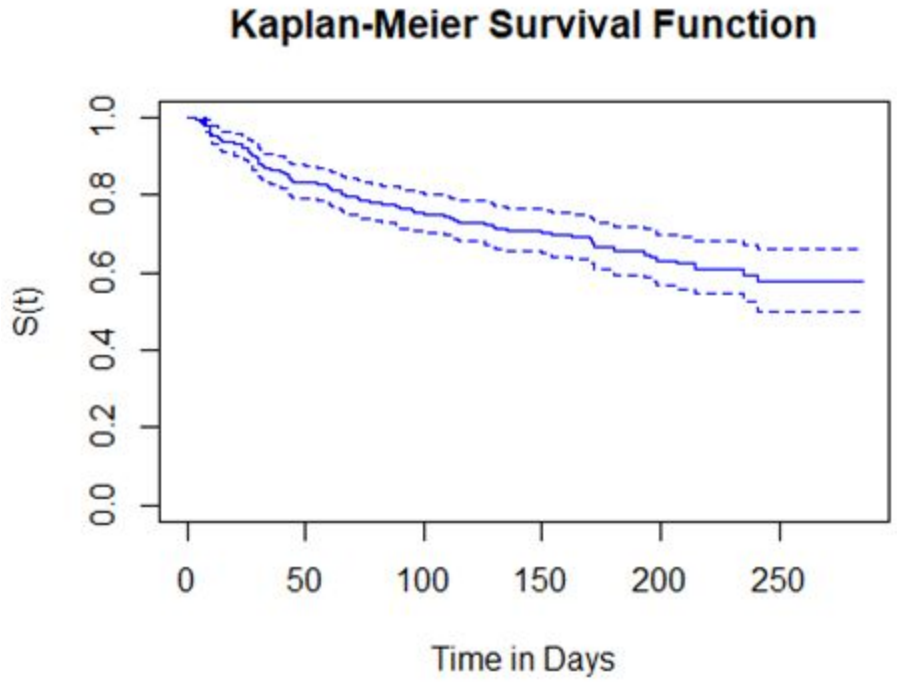
First, we draw Kaplan-Meier plots to both for the entire dataset and categorical variables to have a first glance on potential variables that we want to include in the model. Then, we consider using Cox Proportional Hazard model and Analysis of Variance (ANOVA) Table to determine statistically significant parameters and do model checking to decide the potential models. Next, we consider performing model fitting, i.e. forward selection, to all variables and identify the optimal model by using criteria of finding the smallest AIC or BIC.

With the goal of putting the final model into the practical laboratory/hospital settings, we also want to use machine learning tools to expand more on improving the model. The tools we use are Random Forests and K-Nearest Neighbors**.** By calculating the training accuracy, we will

compare all models and then decide the final model in order to observe the impacts of the existing disease conditions on the survival rate of heart failure patients.

Kaplan-Meier Curves

1. Kaplan-Meier Estimator of the Entire Data Set



(Figure 1)

```
quantile(heart.fit,c(0.25,0.5))$q #survival time at 25th, 50th percentile
```

```
##  25  50
## 100  NA
```

```
summ.fit <- summary(heart.fit)
summ.fit$surv[summ.fit$time == 50] # past 50 days
```

```
## [1] 0.8310352
```

```
summ.fit$surv[summ.fit$time == 100] # past 100 days
```

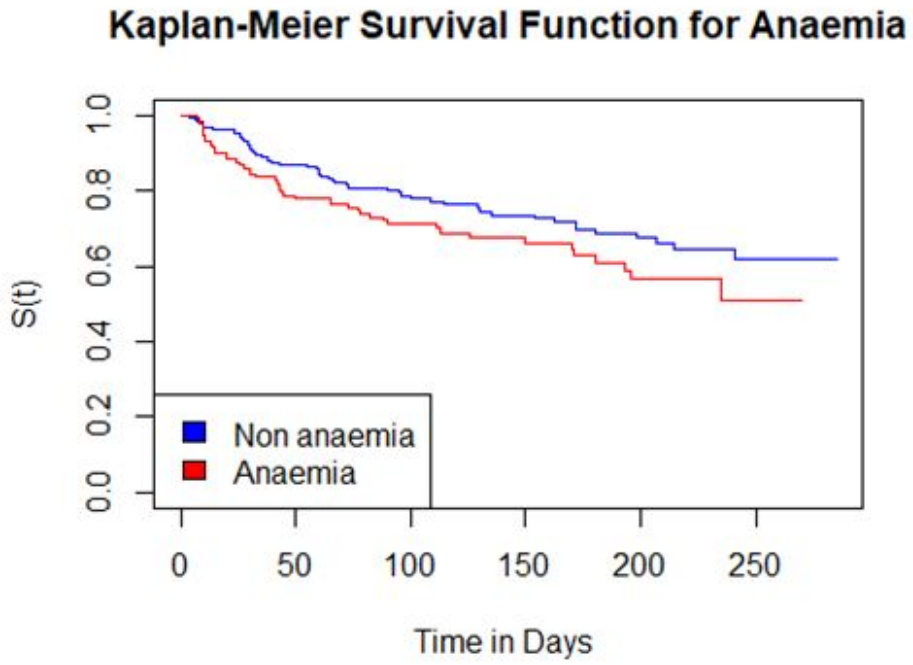
```
## [1] 0.7497495
```

Figure 1 shows the Kaplan–Meier estimate from the data along with a 95% confidence interval. Since the survival function never goes down to 0.5, so there is no median survival rate. While the

survival time at 25th percentile survival rate will be 100. The probability of surviving past 50 days in this sample is 83.10% and the probability of surviving past 100 days in this sample is 74.97%.

2. Kaplan-Meier Estimator for Categorical Features

Among 5 categorical anaemia, high blood pressure, diabetes, sex and smoking, only under the anaemia and high blood pressure factors, there will be a clear differences without intersection for levels.



(Figure 2)

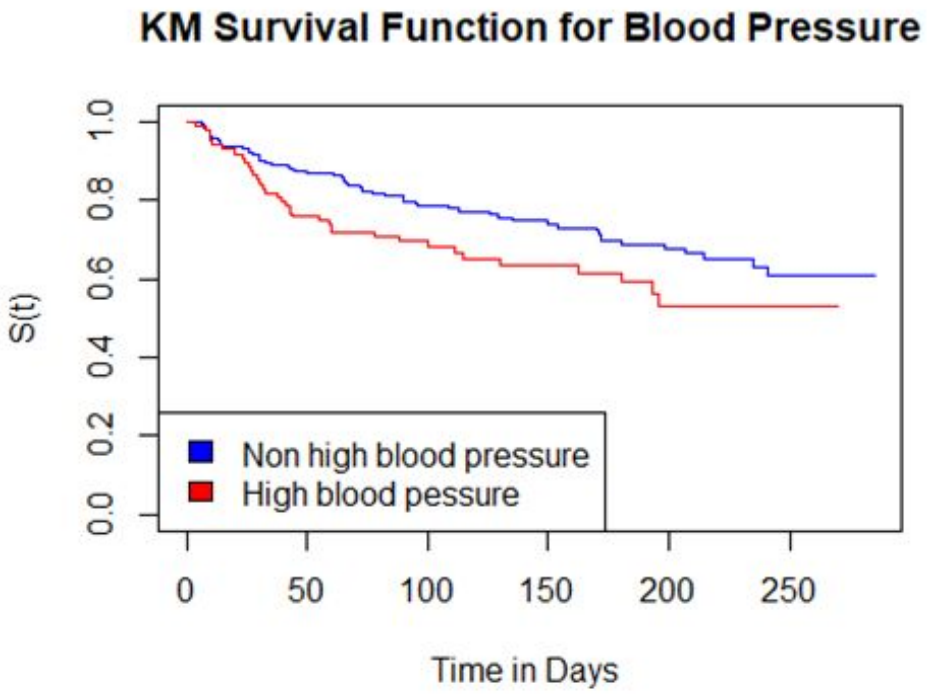
The anaemia has two levels (yes and no). In Figure 2, we could observe that the survival function of non-anaemia is higher than the survival function of anaemia. Besides, the survival time at 25th percentile for non-anaemia is 130, for anaemia is 77. Hence it means the non-anaemia will experience a longer time until failure.

```
quantile(heart.anaemia.fit,0.25)
```

```
## $quantile
##            25
## anaemia=0 130
## anaemia=1  77
##
## $lower
##           25
## anaemia=0 90
## anaemia=1 43
##
## $upper
##            25
## anaemia=0 207
## anaemia=1 170
```

```
quantile(heart.blood.fit,0.25)
```

```
## $quantile
##           25
## blood=0 135
## blood=1  55
##
## $lower
##          25
## blood=0 90
## blood=1 33
##
## $upper
##           25
## blood=0 207
## blood=1 130
```



(Figure 3)

Figure 3 shows that non-high blood pressure has a higher survival function than high blood pressure. Furthermore, the survival time at the 25th percentile for non-high blood pressure is 135, for high blood pressure is 55.

Therefore, we would highly tend to have anaemia and high blood pressure in our model, but in order to have a more insightful model, we will do Cox Proportional model and AIC.

Cox Proportional Hazard Model and Its Checking

In order to construct a complicated model with more factors, we use the Cox Proportional model. Assume we have n factors, the Cox proportional model assumes proportional hazard as: $h_i(t) = h_0(t)\, exp(\beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_n x_{ni})$, where $h_0(t)$ is the baseline hazard function. In short, hazard rate = 1 means no effect, hazard rate < 1 means reduction in the hazard and opposite for hazard rate > 1.

1. Significance of Variables Under Cox PH Assumption

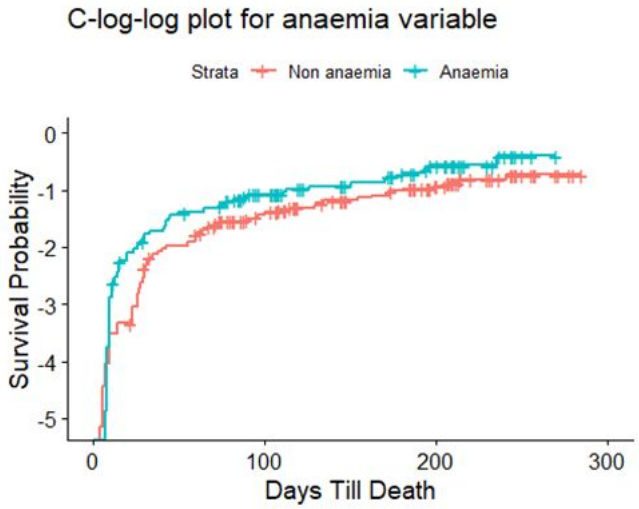
```
## Call:
## coxph(formula = Surv(heart.time, heart.cns) ~ age + creatinine_phosphokinase +
##     anaemia + diabetes + ejection_fraction + high_blood_pressure +
##     platelets + serum_creatinine + serum_sodium + sex + smoking,
##     data = heart)
##
##                               coef  exp(coef)   se(coef)      z        p
## age                       4.641e-02  1.048e+00  9.324e-03  4.977 6.45e-07
## creatinine_phosphokinase  2.207e-04  1.000e+00  9.919e-05  2.225   0.0260
## anaemia                   4.601e-01  1.584e+00  2.168e-01  2.122   0.0338
## diabetes                  1.399e-01  1.150e+00  2.231e-01  0.627   0.5307
## ejection_fraction        -4.894e-02  9.522e-01  1.048e-02 -4.672 2.98e-06
## high_blood_pressure       4.757e-01  1.609e+00  2.162e-01  2.201   0.0278
## platelets                -4.635e-07  1.000e+00  1.126e-06 -0.412   0.6806
## serum_creatinine          3.210e-01  1.379e+00  7.017e-02  4.575 4.76e-06
## serum_sodium             -4.419e-02  9.568e-01  2.327e-02 -1.899   0.0575
## sex                      -2.375e-01  7.886e-01  2.516e-01 -0.944   0.3452
## smoking                   1.289e-01  1.138e+00  2.512e-01  0.513   0.6078
##
```

(Table 1)

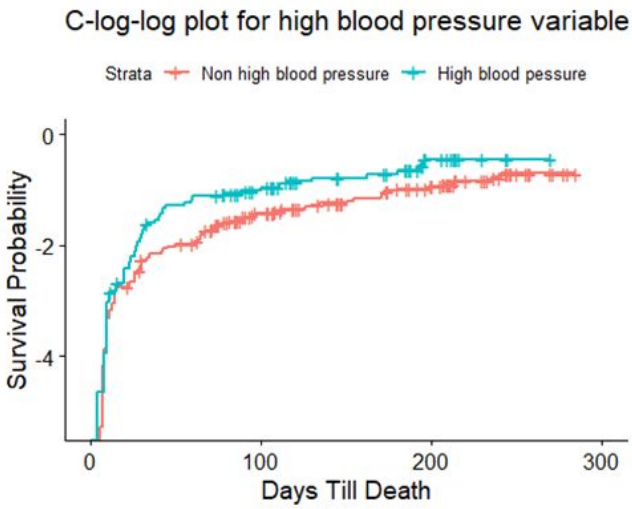
The results of Cox Proportional model are presented in Table 1. The p-value is calculated by coxph function in R. Assume we have 0.5 significance level, skim from all of the p-value, the age, creatinine phosphokinase, anaemia, ejection fraction, high blood pressure and serum creatinine are significant.

2. C log-log Plot and Tests of the Hypothesis

Since the p-value in Table 1 is generated by Cox Proportional Hazard model, we would check if they are meeting the proportional hazards assumption. For two categorical variables, we use the graphical way, and for four numerical variables we apply the test using cox.zph.



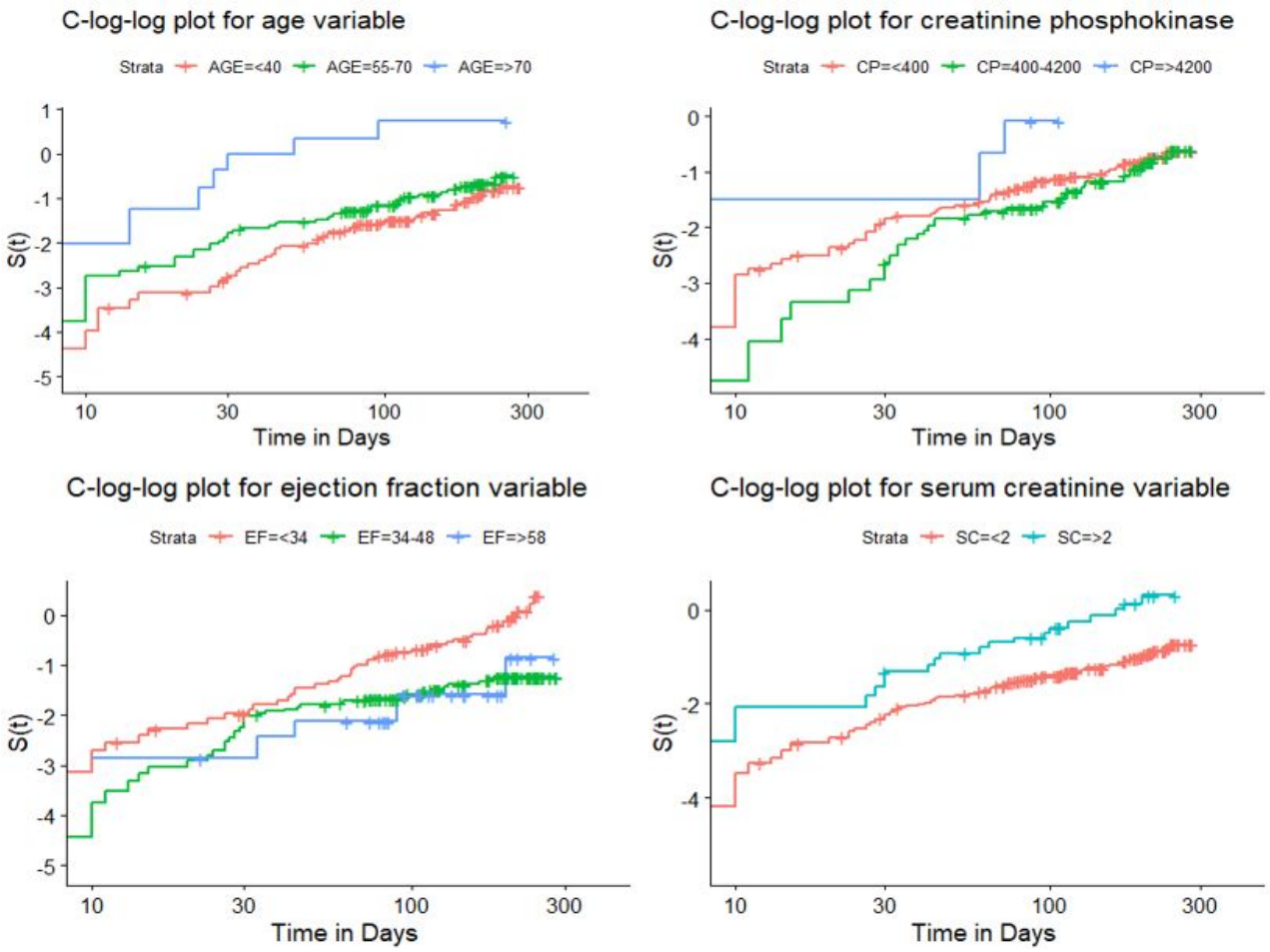
(Figure 4)



(Figure 5)

Figure 4 and Figure 5 are the clog-log plot for anaemia and high blood pressure respectively. There is a little intersection at the beginning, but as time increases, they all diverge. The cox.zph also confirms that anaemia and high blood pressure meet the proportional hazards assumption.

Figure 6 below shows the clog-log plot for four numerical variables: age, creatinine phosphokinase, ejection fraction and serum creatinine. For age and serum creatinine, there are clearly divergences and have a large p-value in the Table 2, so we also assume they meet the proportional hazards assumption. Furthermore, the ejection fraction has p-value less than 0.05 in Table 2 and in the clog-log plot there are few intersections between EF=34-48 and EF>48, therefore the ejection fraction has not met the proportional hazards assumption. Even though creatinine phosphokinase has a large p-value, in the clog-log plot, the CP=400-4200 and CP>4200 looks convergent, so we are a little bit concerned about it and decide not to include it in our model.



(Figure 6)
Clog log plot for Age, CP (Creatinine Phosphokinase), EF(Ejection Fraction) and SC(Serum Creatinine)

```
cox.zph(coxph(Surv(heart.time,heart.cns)~heart$anaemia+heart$high_blood_pressure + heart$age +
                heart$creatinine_phosphokinase +
                heart$ejection_fraction +
                heart$serum_creatinine), global = TRUE)
```

```
##                                chisq df     p
## heart$anaemia                 0.0062  1 0.937
## heart$high_blood_pressure     0.0105  1 0.918
## heart$age                     0.1774  1 0.674
## heart$creatinine_phosphokinase 0.8805  1 0.348
## heart$ejection_fraction       5.0657  1 0.024
## heart$serum_creatinine        1.8459  1 0.174
## GLOBAL                        9.8324  6 0.132
```

(Table 2)

In conclusion, we justify using age, anaemia, high blood pressure, creatinine phosphokinase and serum creatinine meet PH assumption.

ANOVA Table and Its Checking

1. Significance of Variables Under ANOVA Table

Analysis of Variance (ANOVA) table provides the difference among group means in the sample. For choosing the variables, we understand that if the p-value of the variable is smaller than 0.05, then we consider the variable to be statistically significant.

Here, we use code anova(fit) with all variables included to calculate p-value for each variable.

```
#Anova table with all variables
anova(fit)
```

```
## Analysis of Deviance Table
##  Cox model: response is Surv(heart.time, heart.cns)
## Terms added sequentially (first to last)
##
##                            loglik   Chisq Df Pr(>|Chi|)
## NULL                      -509.21
## age                       -497.45 23.5152  1  1.239e-06 ***
## creatinine_phosphokinase -496.79  1.3243  1    0.24983
## anaemia                   -495.47  2.6374  1    0.10438
## diabetes                  -495.30  0.3274  1    0.56717
## ejection_fraction         -482.59 25.4242  1  4.601e-07 ***
## high_blood_pressure       -480.07  5.0497  1    0.02463 *
## platelets                 -479.80  0.5373  1    0.46357
## serum_creatinine          -470.28 19.0443  1  1.277e-05 ***
## serum_sodium              -468.68  3.1894  1    0.07412 .
## sex                       -468.36  0.6434  1    0.42250
## smoking                   -468.23  0.2620  1    0.60877
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Table 3)

The above table is the summary of the ANOVA table. From the table, we can observe that for variable age, ejection fraction, high blood pressure, and serum creatinine, the p-value is smaller than 0.05, i.e. statistically significant. Therefore, we choose these four variables: age, ejection fraction, high blood pressure, and serum creatinine to be included in the model.

2. Checking PH Assumption

For checking PH assumptions, we mainly use the test of the hypothesis on PH model which is cox.zph in R code and graphical method which is the C Log-Log plots to help us identify whether the model is correct for proportional hazard assumptions or not.

From the ANOVA table, we select four significant variables: age, ejection fraction, high blood pressure and serum creatinine. Then, we work on checking the PH assumption of the model with these four variables. The below table is the result for the cox.zph on the model.

```
#fit model
cox2 = coxph(Surv(heart.time,heart.cns) ~ age +ejection_fraction +
high_blood_pressure+ serum_creatinine,
data=heart)

#Check PH assumption
cox.zph(cox2)
```

```
##                     chisq df     p
## age                 0.271  1 0.603
## ejection_fraction   4.394  1 0.036
## high_blood_pressure 0.034  1 0.854
## serum_creatinine    1.289  1 0.256
## GLOBAL              6.304  4 0.178
```

(Table 4)

We know that if all p-values are larger than 0.05, then we can conclude that PH assumption is correct. However, since the p-value for variable ejection fraction is 0.036, smaller than 0.05, then this variable violates the cox proportional hazard model.

Model Fitting

1. Forward Selection with Comparing AIC/BIC Value

For the model fitting part, we follow the forward selection and check the AIC values. First, we fit each covariate separately, and then find the model with covariate(s) which has the smallest AIC. After this, we keep the covariate(s), add one additional covariate in the model and repeat the above procedures.

Single variable model

```
#AIC
model1 = coxph(Surv(heart.time,heart.cns) ~ age,data=heart)
model2 = coxph(Surv(heart.time,heart.cns) ~ anaemia,data=heart)
model3 = coxph(Surv(heart.time,heart.cns) ~ creatinine_phosphokinase,data=heart)
model4 = coxph(Surv(heart.time,heart.cns) ~ diabetes,data=heart)
model5 = coxph(Surv(heart.time,heart.cns) ~ ejection_fraction,data=heart)
model6 = coxph(Surv(heart.time,heart.cns) ~ high_blood_pressure,data=heart)
model7 = coxph(Surv(heart.time,heart.cns) ~ platelets,data=heart)
model8 = coxph(Surv(heart.time,heart.cns) ~ serum_creatinine,data=heart)
model9 = coxph(Surv(heart.time,heart.cns) ~ serum_sodium,data=heart)
model10 = coxph(Surv(heart.time,heart.cns) ~ sex,data=heart)
model11 = coxph(Surv(heart.time,heart.cns) ~ smoking,data=heart)
```

By creating models which only contain single variable, we perform AIC on each of them as the following:

```
AIC(model1,model2,model3,model4,model5,model6,model7,model8)
```

```
##        df       AIC
## model1  1  996.8950
## model2  1 1017.7259
## model3  1 1019.3056
## model4  1 1020.3694
## model5  1  999.8589
## model6  1 1016.2152
## model7  1 1019.8608
## model8  1 1002.4730
```

Tidying up the above models with their corresponding AIC value, we can get the following result:

Model with variable	AIC value
age	**996.8950**
anaemia	1017.7259
creatinine phosphokinase	1019.3056
diabetes	1020.3694
ejection fraction	999.8589
high blood pressure	1016.2152
platelets	1019.8608
serum creatinine	1002.4730
serum sodium	1010.1813
sex	1020.4062
smoking	1020.4084

(Table 5)

From the above table, we pick the age variable since it has the smallest AIC value.

Model with one additional variable after picking Age

Similar to the above step, we first add age into each model and fit models with one additional variable as the following:

```
#fit a second variable
model2.1 = coxph(Surv(heart.time,heart.cns) ~ age+anaemia,data=heart)
model2.2 = coxph(Surv(heart.time,heart.cns) ~ age+creatinine_phosphokinase,data=heart)
model2.3 = coxph(Surv(heart.time,heart.cns) ~ age+diabetes,data=heart)
model2.5 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction,data=heart) #smallest AIC
model2.6 = coxph(Surv(heart.time,heart.cns) ~ age+high_blood_pressure,data=heart)
model2.7 = coxph(Surv(heart.time,heart.cns) ~ age+platelets,data=heart)
model2.8 = coxph(Surv(heart.time,heart.cns) ~ age+serum_creatinine,data=heart)
model2.9 = coxph(Surv(heart.time,heart.cns) ~ age+serum_sodium,data=heart)
model2.10 = coxph(Surv(heart.time,heart.cns) ~ age+sex,data=heart)
model2.11 = coxph(Surv(heart.time,heart.cns) ~ age+smoking,data=heart)

AIC(model2.1,model2.2,model2.3,model2.5,model2.6,model2.7,model2.8,model2.9,model2.10,model2.11)
```

```
##           df      AIC
## model2.1   2 996.9942
## model2.2   2 997.5708
## model2.3   2 998.5990
## model2.5   2 974.1119
## model2.6   2 995.0472
## model2.7   2 998.3989
## model2.8   2 981.9514
## model2.9   2 988.8779
## model2.10  2 998.8914
## model2.11  2 998.8563
```

Tidying up the above models with their corresponding AIC value, we can get the following result:

Model with variables	AIC value
age + anaemia	996.9942
age + creatinine phosphokinase	997.5708
age + diabetes	998.5990
age + ejection fraction	**974.1119**
age + high blood pressure	995.0472
age + platelets	998.3989
age + serum creatinine	981.9514
age + serum sodium	998.8770
age + sex	998.8914
age + smoking	998.8563

(Table 6)

From the above table, we can observe that model with variable age and ejection fraction has the smallest value 974.1119. Also, compared to the AIC value of the previous model with only one variable age, this model has a smaller AIC value. Therefore, we can include variables age and ejection fraction in the model.

Repeating the above procedures and after several steps, we find the model with variables age, ejection fraction, serum creatinine, high blood pressure, anaemia, creatinine phosphokinase, and serum sodium has the least AIC value (**951.8277**). Therefore, we include these variables in our model from the AIC forward selection method. For the entire model building with AIC values, please refer to the appendix.

Since the model we have from picking the smallest AIC value includes too many parameters, we would like to perform BIC value as well in order to keep our model simple.

df <dbl>	BIC <dbl>
1	999.4594
1	1020.2903
1	1021.8700
1	1022.9338
1	1002.4233
1	1018.7795
1	1022.4251
1	1005.0373

BIC values of single variable models

df <dbl>	BIC <dbl>
2	1002.1229
2	1002.6995
2	1003.7277
2	979.2406
2	1000.1759
2	1003.5276
2	987.0801
2	994.0066
2	1004.0201
2	1003.9850

BIC values of two variables models

df <dbl>	BIC <dbl>
3	980.8977
3	982.8093
3	983.6289
3	978.7571
3	983.6235
3	965.5937
3	977.9353
3	983.1547
3	983.7276

BIC values of three variables models

df <dbl>	BIC <dbl>
4	967.1638
4	968.2955
4	969.3364
4	965.3614
4	970.1521
4	968.1639
4	969.5882
4	970.1574

BIC values of four variables models

df <dbl>	BIC <dbl>
5	967.4021
5	967.5976
5	969.2471
5	969.8634
5	967.4812
5	969.6177
5	969.9235

BIC values of five variables models

By looking at the BIC values which take the number of parameters into consideration, we find the model *Surv ~ age+ejection fraction+serum creatinine+high blood pressure* has the least BIC value (**965.3614**). This model is the same as the model built under the ANOVA table. Therefore, we will include this model as one of the proposed models.

2. Interaction Term

Furthermore, we also consider cases which models with potential interaction terms based on confounding variables: age*serum creatinine, ejection fraction*serum creatinine, serum creatinine*high blood pressure, and anaemia*high blood pressure.

By creating models with above interaction terms, we perform AIC on each of them as the following:

```
#Interaction term
fit.across1<-  coxph(Surv(heart.time,heart.cns) ~ age*serum_creatinine,data=heart)
fit.across2<-  coxph(Surv(heart.time,heart.cns) ~ ejection_fraction*serum_creatinine,data=heart)
fit.across3<-coxph(Surv(heart.time,heart.cns) ~ serum_creatinine*high_blood_pressure,data=heart)
fit.across4<-coxph(Surv(heart.time,heart.cns) ~ anaemia*high_blood_pressure,data=heart)
AIC(fit.across1,fit.across2,fit.across3,fit.across4)
```

```
##             df       AIC
## fit.across1  3  982.5491
## fit.across2  3  981.4386
## fit.across3  3 1001.6256
## fit.across4  3 1017.5627
```

Tidying up the above models with their corresponding AIC value, we can get the following result:

Model with Interaction Term	AIC Value
age * serum creatinine	982.5491
ejection fraction * serum creatinine	**981.4386**
serum creatinine * high blood pressure	1001.6256
anaemia * high blood pressure	1017.5627

(Table 7)

The above table shows that none of these interaction terms get a smaller AIC value than the models mentioned above in the previous section (**951.8277**). Therefore, we decide not to use interaction terms in our model.

Proposed Models

1. Model 1 (from Cox Proportional Hazard model):
   Surv ~ age+anaemia+high blood pressure+serum creatinine

2. Model 2 (from Anova Table and BIC):
   Surv ~ age+ejection fraction+serum creatinine+high blood pressure

Extension 1 - Stratification on Ejection Fraction

In the earlier section "Cox Proportional Hazard Model and Its Checking", we notice that variable ejection fraction fails under proportional hazards assumption, but in ANOVA and BIC section we found that ejection fraction is significant and could give us a decent result. Therefore, we would like to build a stratification on ejection fraction.

```
#Stratification
model_strata = coxph(Surv(heart.time,heart.cns)~ age +
                       strata(ejection_fraction) +
                       high_blood_pressure +
                       serum_creatinine,data=heart)
model_strata
```

```
## Call:
## coxph(formula = Surv(heart.time, heart.cns) ~ age + strata(ejection_fraction) +
##     high_blood_pressure + serum_creatinine, data = heart)
##
##                         coef exp(coef) se(coef)     z        p
## age                 0.046824  1.047938 0.009816 4.770 1.84e-06
## high_blood_pressure 0.655838  1.926757 0.229162 2.862 0.004211
## serum_creatinine    0.339955  1.404884 0.102761 3.308 0.000939
##
## Likelihood ratio test=44.17  on 3 df, p=1.391e-09
## n= 299, number of events= 96
```

The p-value of likelihood ratio test is very small (less than 0.05), besides age, high blood pressure and serum creatinine are also significant in the stratification model, hence we would reject null and claim that high blood pressure and serum creatinine are statistically significant on the time until relapse.

Extension 2 - Model Improvement using Machine Learning

After we proposed some possible models, we would like to use machine learning tools to further build classifiers. The goal of doing this is not only to see what are the most important risk factors, but also to see if our potential finding could have clinical practice in prediction, that is to say, the physician and the doctors may only need to see a few important indicators to determine if a heart failure patient will survive or not.

The method of doing machine learning is to treat the status as an indicator, and we also include each patient's follow-up time in our machine learning model since including four variables from the candidate model2 will somehow ignore the variable time, but survival time is also a potential parameter for building the classifier. We split data into 75% training set and 25% test set, which gives us 224 observations in the training set and 75 observations in the test set. For the K-Nearest

Neighbors algorithm, the best fold is 10. We obtain our training and testing accuracy as well as error for both models. Since our testing accuracy is slightly smaller than training accuracy, we can conclude the training and test sets were randomly selected.

Below is the code for model *Surv ~ age+ejection fraction+serum creatinine+high blood pressure,* the same code for another model with only changing the variables inside (see appendix for more details). The final result of accuracy is shown below in Table 8 below.

```
# KNN
best.kfold <- 10
Xtest <- heart_test %>%
  dplyr::select(age, anaemia, high_blood_pressure,
                serum_creatinine, time) %>%
  as.matrix()

# Traning data outcome
Ytest <- heart_test$y

set.seed(1)
## get classifications for training set
predYtr <- knn(train = Xdat, test = Xdat, cl = Ydat, k = best.kfold)

## get classifications for testing set
predYvl <- knn(train = Xdat, test = Xtest, cl = Ydat, k = best.kfold)

#records[1, 1] <- calc_error_rate(predYtr, Ydat)
records[1, 1] <- (1-calc_error_rate(predYvl, Ytest))
```

```
# Random Forest
heart.rf <- randomForest( y ~  age + anaemia + high_blood_pressure
                        + serum_creatinine + time,
                    data =  heart_train,importance=TRUE)
rf.predict<-predict(heart.rf,newdata=heart_test)
```

```
## random forest confusion matrix
rf.err = table(pred = rf.predict, truth = heart_test$y)
test.rf.err = 1 - sum(diag(rf.err))/sum(rf.err)
records[2, 1]=1- test.rf.err
```

```
records
```

```
##                Accuracy
## knn           0.7600000
## random forest 0.7733333
```

Model	KNN	Random Forest
age, anaemia, high blood pressure, serum creatinine, time	0.7600	0.7733
age, ejection fraction, high blood pressure, serum creatinine, time	0.8267	0.8133

(Table 8)

Overall, the second model with **age, ejection fraction, high blood pressure, serum creatinine and time** gives us a better result, especially the KNN algorithm. Hence, in a practical sense, if a physician is given those clinical features, there is a 82.67% chance for them to know if the patient will survive or not by using the second model and knowing information on age, ejection fraction, high blood pressure, serum creatinine, and time. Therefore, we choose the final survival analysis model to be *Surv ~ age+ejection fraction+serum creatinine+high blood pressure.*

Hazard Ratio and Confidence Interval

We use exp(confint( )) to calculate the 95% confidence interval for each variable in final model:

```
model2 = coxph(Surv(heart.time,heart.cns) ~ age + ejection_fraction +
                 serum_creatinine + high_blood_pressure,
               data = heart)
exp(confint(model2,level = 0.95))
```

	exp(coef)	exp(-coef)	lower .95	upper .95
age	1.0452	0.9568	1.0268	1.0638
ejection_fraction	0.9516	1.0508	0.9332	0.9704
serum_creatinine	1.4148	0.7068	1.2414	1.6124
high_blood_pressure	1.6020	0.6242	1.0585	2.4244

From the result above, we know that:

the hazard ratio for age is 1.0452, and its 95% CI is [1.0268, 1.0638];
the hazard ratio for ejection fraction is 1.0452, and its 95% CI is [0.9332, 0.9704];
the hazard ratio for serum creatinine is 1.4148, and its 95% CI is [1.2414,1.6124];
the hazard ratio for high blood pressure is 1.6020, and its 95% CI is [1.0585, 2.4244].

All of the hazard ratios are greater than 1, and none of the confidence intervals include the null hypothesis that the ratio is 1. The confidence intervals are fairly narrow, which means that our hazard ratio estimates are precise.

Conclusion

From our analyses of the dataset using survival analysis tools, we found the best model to use to predict heart failure patients' survival rate is *Surv ~ age+ejection fraction+serum creatinine+high blood pressure*. Therefore, we conclude that clinical features age, anaemia, ejection fraction, high blood pressure and serum creatinine have significant impacts on heart failure patients' survival rate. Moreover, the result from machine learning suggests that age, ejection fraction, high blood pressure and serum creatinine are potential useful features to predict if a heart failure patient will survive or not.

References

1. Ahmad T, Munir A, Bhatti SH, Aftab M, Raza MA (2017) Survival analysis of heart failure patients: A case study. PLoS ONE 12(7): e0181001. https://doi.org/10.1371/journal.pone.0181001
2. Chicco and Jurman BMC Medical Informatics and Decision Making (2020) 20:16 https://doi.org/10.1186/s12911-020-1023-5
3. UCI Machine Learning Repository, https://archive.ics.uci.edu/ml/datasets/Heart+failure+clinical+records

Appendix

```r
#setwd('C:/Users/Julia/Box/pstat175/project')
heart = read.csv("heart_failure_clinical_records_dataset.csv",
                 header = TRUE)

heart.time=as.vector(heart$time)
heart.cns=as.vector(heart$DEATH_EVENT)
heart.fit = surv_fit(Surv(heart.time,heart.cns)~1, data=heart)
plot(heart.fit,
     main = "Kaplan-Meier Survival Function",
     xlab = "Time in Days",
     ylab = "S(t)",
     col = "blue")

sex = as.factor(heart$sex)
heart.sex.fit = surv_fit(Surv(heart.time,heart.cns)~sex,
                         data=heart)
plot(heart.sex.fit,
     main = "Kaplan-Meier Survival Function for gender",
     xlab = "Time in Days ",
     ylab = "S(t)",
     col= c("blue","red"))
legend("bottomleft",legend = c("Female","Male"),fill=c("blue","red"))

anaemia = as.factor(heart$anaemia)
heart.anaemia.fit = surv_fit(Surv(heart.time,heart.cns)~anaemia,
                         data=heart)
plot(heart.anaemia.fit,
     main = "Kaplan-Meier Survival Function for Anaemia",
     xlab = "Time in Days ",
     ylab = "S(t)",
     col= c("blue","red"))
legend("bottomleft",legend = c("Non anaemia","Anaemia"),fill=c("blue","red"))

smoking = as.factor(heart$smoking)
heart.smoking.fit = surv_fit(Surv(heart.time,heart.cns)~smoking,
                         data=heart)
plot(heart.smoking.fit,
     main = "Kaplan-Meier Survival Function for smoking",
     xlab = "Time in Days ",
     ylab = "S(t)",
     col= c("blue","red"))
legend("bottomleft",legend = c("Non smoking","Smoking"),fill=c("blue","red"))

diabetes = as.factor(heart$diabetes)
heart.diabetes.fit = surv_fit(Surv(heart.time,heart.cns)~diabetes,
                         data=heart)
plot(heart.diabetes.fit,
     main = "Kaplan-Meier Survival Function for diabetes",
     xlab = "Time in Days ",
     ylab = "S(t)",
     col= c("blue","red"))
legend("bottomleft",legend = c("Non diabete","diabete"),fill=c("blue","red"))

blood = as.factor(heart$high_blood_pressure)
heart.blood.fit = surv_fit(Surv(heart.time,heart.cns)~blood,
                         data=heart)
plot(heart.blood.fit,
     main = "Kaplan-Meier Survival Function for Blood Pressure",
     xlab = "Time in Days ",
     ylab = "S(t)",
     col= c("blue","red"))
legend("bottomleft",legend = c("Non high blood pressure","High blood pessure"),fill=c("blue","red"))

summ.fit <- summary(heart.fit)
summ.fit$surv[summ.fit$time == 50] # past 50 days

## [1] 0.8310352

summ.fit$surv[summ.fit$time == 100] # past 100 days

## [1] 0.7497495
```

```r
heart.anaemia.fit = survfit(Surv(heart.time,heart.cns)~anaemia,
                            data=heart)
quantile(heart.anaemia.fit,0.25)
```

```
## $quantile
##            25
## anaemia=0 130
## anaemia=1  77
##
## $lower
##           25
## anaemia=0 90
## anaemia=1 43
##
## $upper
##            25
## anaemia=0 207
## anaemia=1 170
```

```r
heart.blood.fit = survfit(Surv(heart.time,heart.cns)~blood,
                            data=heart)
quantile(heart.blood.fit,0.25)
```

```
## $quantile
##          25
## blood=0 135
## blood=1  55
##
## $lower
##         25
## blood=0 90
## blood=1 33
##
## $upper
##          25
## blood=0 207
## blood=1 130
```

```r
# fit for all variables
fit = coxph(Surv(heart.time,heart.cns) ~ age +
              creatinine_phosphokinase + anaemia + diabetes +
              ejection_fraction + high_blood_pressure +
              platelets + serum_creatinine + serum_sodium +
              sex + smoking, heart)
fit
```

```
## Call:
## coxph(formula = Surv(heart.time, heart.cns) ~ age + creatinine_phosphokinase +
##     anaemia + diabetes + ejection_fraction + high_blood_pressure +
##     platelets + serum_creatinine + serum_sodium + sex + smoking,
##     data = heart)
##
##                               coef  exp(coef)   se(coef)      z        p
## age                       4.641e-02  1.048e+00  9.324e-03  4.977 6.45e-07
## creatinine_phosphokinase  2.207e-04  1.000e+00  9.919e-05  2.225   0.0260
## anaemia                   4.601e-01  1.584e+00  2.168e-01  2.122   0.0338
## diabetes                  1.399e-01  1.150e+00  2.231e-01  0.627   0.5307
## ejection_fraction        -4.894e-02  9.522e-01  1.048e-02 -4.672 2.98e-06
## high_blood_pressure       4.757e-01  1.609e+00  2.162e-01  2.201   0.0278
## platelets                -4.635e-07  1.000e+00  1.126e-06 -0.412   0.6806
## serum_creatinine          3.210e-01  1.379e+00  7.017e-02  4.575 4.76e-06
## serum_sodium             -4.419e-02  9.568e-01  2.327e-02 -1.899   0.0575
## sex                      -2.375e-01  7.886e-01  2.516e-01 -0.944   0.3452
## smoking                   1.289e-01  1.138e+00  2.512e-01  0.513   0.6078
##
## Likelihood ratio test=81.95  on 11 df, p=6.173e-13
## n= 299, number of events= 96
```

```r
#Anova table with all variables
anova(fit)
```

```
## Analysis of Deviance Table
##  Cox model: response is Surv(heart.time, heart.cns)
## Terms added sequentially (first to last)
##
##                           loglik   Chisq Df Pr(>|Chi|)
```

```
## NULL                       -509.21
## age                        -497.45 23.5152  1  1.239e-06 ***
## creatinine_phosphokinase -496.79  1.3243  1    0.24983
## anaemia                    -495.47  2.6374  1    0.10438
## diabetes                   -495.30  0.3274  1    0.56717
## ejection_fraction          -482.59 25.4242  1  4.601e-07 ***
## high_blood_pressure        -480.07  5.0497  1    0.02463 *
## platelets                  -479.80  0.5373  1    0.46357
## serum_creatinine           -470.28 19.0443  1  1.277e-05 ***
## serum_sodium               -468.68  3.1894  1    0.07412 .
## sex                        -468.36  0.6434  1    0.42250
## smoking                    -468.23  0.2620  1    0.60877
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the ANOVA table, the significant variables are age, ejection_fraction, high_blood_pressure and serum_creatinine.

```
#fit model2
cox2 = coxph(Surv(heart.time,heart.cns) ~ age +ejection_fraction +
high_blood_pressure+ serum_creatinine,
data=heart)
cox2

## Call:
## coxph(formula = Surv(heart.time, heart.cns) ~ age + ejection_fraction +
##     high_blood_pressure + serum_creatinine, data = heart)
##
##                           coef exp(coef)  se(coef)      z        p
## age                   0.044173  1.045163  0.009027  4.894 9.91e-07
## ejection_fraction    -0.049587  0.951623  0.009968 -4.975 6.54e-07
## high_blood_pressure   0.471236  1.601973  0.211410  2.229   0.0258
## serum_creatinine      0.347001  1.414818  0.066705  5.202 1.97e-07
##
## Likelihood ratio test=71.31  on 4 df, p=1.203e-14
## n= 299, number of events= 96

#Check PH assumption
cox.zph(cox2)

##                      chisq df     p
## age                  0.271  1 0.603
## ejection_fraction    4.394  1 0.036
## high_blood_pressure 0.034  1 0.854
## serum_creatinine     1.289  1 0.256
## GLOBAL               6.304  4 0.178

#fit model
cox1 = coxph(Surv(heart.time,heart.cns) ~ age +ejection_fraction +
high_blood_pressure+ serum_creatinine+creatinine_phosphokinase+anaemia,
data=heart)

#Check PH assumption
cox.zph(cox1)

##                           chisq df     p
## age                      0.1774  1 0.674
## ejection_fraction        5.0657  1 0.024
## high_blood_pressure      0.0105  1 0.918
## serum_creatinine         1.8459  1 0.174
## creatinine_phosphokinase 0.8805  1 0.348
## anaemia                  0.0062  1 0.937
## GLOBAL                   9.8324  6 0.132
```

Ejection_fraction failed the ph assumption test.

```
#cloglog plots
cloglog<- function(x){log(-log(x))}
ggsurvplot(heart.sex.fit, fun=cloglog,
           legend.labs=c("Female","Male"))+labs(x="Days Till Death",y="Survival Probability",title ="C-log-log plot for se
x variable")

ggsurvplot(heart.anaemia.fit, fun=cloglog,
           legend.labs=c("Non anaemia","Anaemia"))+labs(x="Days Till Death",y="Survival Probability",title="C-log-log plot
 for anaemia variable")
```

```r
ggsurvplot(heart.smoking.fit, fun=cloglog,
           legend.labs=c("Non smoking","Smoking"))+labs(x="Days Till Death",y="Survival Probability",title="C-log-log plot
 for smoking variable")

ggsurvplot(heart.blood.fit, fun=cloglog,
           legend.labs=c("Non high blood pressure","High blood pessure"))+labs(x="Days Till Death",y="Survival Probability
",title="C-log-log plot for high blood pressure variable")

AGE = as.factor(ceiling((heart$age-10)/25))
levels(AGE) <- c("<40","55-70",">70")
ggsurvplot(survfit(Surv(heart.time,heart.cns)~ AGE,
                   data=heart),fun="cloglog",xlim=c(10,400),
           xlab = "Time in Days ",
           ylab = "S(t)",title="C-log-log plot for age variable")

CP = as.factor(ceiling((heart$creatinine_phosphokinase-400)/3800))
levels(CP) <- c("<400","400-4200",">4200")
ggsurvplot(survfit(Surv(heart.time,heart.cns)~ CP,
                   data=heart),fun="cloglog",xlim=c(10,400),
           xlab = "Time in Days ",
           ylab = "S(t)",title="C-log-log plot for creatinine phosphokinase")

EF = as.factor(ceiling((heart$ejection_fraction-10)/24))
levels(EF) <- c("<34","34-48",">58")
ggsurvplot(survfit(Surv(heart.time,heart.cns)~ EF,
                   data=heart),fun="cloglog",xlim=c(10,400),
           xlab = "Time in Days ",
           ylab = "S(t)",title="C-log-log plot for ejection fraction variable")

SC = as.factor(ceiling((heart$serum_creatinine-2)/8))
levels(SC) <- c("<2",">2")
ggsurvplot(survfit(Surv(heart.time,heart.cns)~ SC,
                   data=heart),fun="cloglog",xlim=c(10,400),
           xlab = "Time in Days ",
           ylab = "S(t)",title="C-log-log plot for serum creatinine variable")

cox.zph(coxph(Surv(heart.time,heart.cns)~heart$anaemia+heart$high_blood_pressure + heart$age +
                heart$creatinine_phosphokinase +
                heart$ejection_fraction +
                heart$serum_creatinine), global = TRUE)
```

```
##                                chisq df     p
## heart$anaemia                 0.0062  1 0.937
## heart$high_blood_pressure     0.0105  1 0.918
## heart$age                     0.1774  1 0.674
## heart$creatinine_phosphokinase 0.8805  1 0.348
## heart$ejection_fraction       5.0657  1 0.024
## heart$serum_creatinine        1.8459  1 0.174
## GLOBAL                        9.8324  6 0.132
```

```r
#AIC
model1 = coxph(Surv(heart.time,heart.cns) ~ age,data=heart)
model2 = coxph(Surv(heart.time,heart.cns) ~ anaemia,data=heart)
model3 = coxph(Surv(heart.time,heart.cns) ~ creatinine_phosphokinase,data=heart)
model4 = coxph(Surv(heart.time,heart.cns) ~ diabetes,data=heart)
model5 = coxph(Surv(heart.time,heart.cns) ~ ejection_fraction,data=heart)
model6 = coxph(Surv(heart.time,heart.cns) ~ high_blood_pressure,data=heart)
model7 = coxph(Surv(heart.time,heart.cns) ~ platelets,data=heart)
model8 = coxph(Surv(heart.time,heart.cns) ~ serum_creatinine,data=heart)
model9 = coxph(Surv(heart.time,heart.cns) ~ serum_sodium,data=heart)
model10 = coxph(Surv(heart.time,heart.cns) ~ sex,data=heart)
model11 = coxph(Surv(heart.time,heart.cns) ~ smoking,data=heart)

AIC(model1,model2,model3,model4,model5,model6,model7,model8)
```

```
##        df       AIC
## model1  1  996.8950
## model2  1 1017.7259
## model3  1 1019.3056
## model4  1 1020.3694
## model5  1  999.8589
## model6  1 1016.2152
## model7  1 1019.8608
## model8  1 1002.4730
```

```r
BIC(model1,model2,model3,model4,model5,model6,model7,model8)
```

```
##        df       BIC
## model1  1  999.4594
## model2  1 1020.2903
## model3  1 1021.8700
## model4  1 1022.9338
## model5  1 1002.4233
## model6  1 1018.7795
## model7  1 1022.4251
## model8  1 1005.0373
```

Age is our pick.

```
#fit a second variable
model2.1 = coxph(Surv(heart.time,heart.cns) ~ age+anaemia,data=heart)
model2.2 = coxph(Surv(heart.time,heart.cns) ~ age+creatinine_phosphokinase,data=heart)
model2.3 = coxph(Surv(heart.time,heart.cns) ~ age+diabetes,data=heart)
model2.5 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction,data=heart) #smallest AIC
model2.6 = coxph(Surv(heart.time,heart.cns) ~ age+high_blood_pressure,data=heart)
model2.7 = coxph(Surv(heart.time,heart.cns) ~ age+platelets,data=heart)
model2.8 = coxph(Surv(heart.time,heart.cns) ~ age+serum_creatinine,data=heart)
model2.9 = coxph(Surv(heart.time,heart.cns) ~ age+serum_sodium,data=heart)
model2.10 = coxph(Surv(heart.time,heart.cns) ~ age+sex,data=heart)
model2.11 = coxph(Surv(heart.time,heart.cns) ~ age+smoking,data=heart)

AIC(model2.1,model2.2,model2.3,model2.5,model2.6,model2.7,model2.8,model2.9,model2.10,model2.11)
```

```
##           df      AIC
## model2.1   2 996.9942
## model2.2   2 997.5708
## model2.3   2 998.5990
## model2.5   2 974.1119
## model2.6   2 995.0472
## model2.7   2 998.3989
## model2.8   2 981.9514
## model2.9   2 988.8779
## model2.10  2 998.8914
## model2.11  2 998.8563
```

```
BIC(model2.1,model2.2,model2.3,model2.5,model2.6,model2.7,model2.8,model2.9,model2.10,model2.11)
```

```
##           df       BIC
## model2.1   2 1002.1229
## model2.2   2 1002.6995
## model2.3   2 1003.7277
## model2.5   2  979.2406
## model2.6   2 1000.1759
## model2.7   2 1003.5276
## model2.8   2  987.0801
## model2.9   2  994.0066
## model2.10  2 1004.0201
## model2.11  2 1003.9850
```

```
#fit a third variable
model3.1 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+anaemia,data=heart)
model3.2 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+creatinine_phosphokinase,data=heart)
model3.3 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+diabetes,data=heart)
model3.6 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+high_blood_pressure,data=heart)
model3.7 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+platelets,data=heart)
model3.8 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine,data=heart)#smallest AIC
model3.9 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_sodium,data=heart)
model3.10 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+sex,data=heart)
model3.11 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+smoking,data=heart)


AIC(model3.1,model3.2,model3.3,model3.6,model3.7,model3.8,model3.9,model3.10,model3.11)
```

```
##           df      AIC
## model3.1   3 973.2047
## model3.2   3 975.1163
## model3.3   3 975.9358
## model3.6   3 971.0641
## model3.7   3 975.9305
## model3.8   3 957.9006
## model3.9   3 970.2423
## model3.10  3 975.4616
## model3.11  3 976.0345
```

```
BIC(model3.1,model3.2,model3.3,model3.6,model3.7,model3.8,model3.9,model3.10,model3.11)
```

```
##           df      BIC
## model3.1   3 980.8977
## model3.2   3 982.8093
## model3.3   3 983.6289
## model3.6   3 978.7571
## model3.7   3 983.6235
## model3.8   3 965.5937
## model3.9   3 977.9353
## model3.10  3 983.1547
## model3.11  3 983.7276
```

```
#fit a 4th variable
model4.1 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+anaemia,data=heart)
model4.2 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+creatinine_phosphokinase,data=heart)
model4.3 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+diabetes,data=heart)
model4.6 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+high_blood_pressure,data=heart)#smalle
st AIC
model4.7 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+platelets,data=heart)
model4.9 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+serum_sodium,data=heart)
model4.10 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+sex,data=heart)
model4.11 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+smoking,data=heart)

AIC(model4.1,model4.2,model4.3,model4.6,model4.7,model4.9,model4.10,model4.11)
```

```
##           df      AIC
## model4.1   4 956.9064
## model4.2   4 958.0381
## model4.3   4 959.0790
## model4.6   4 955.1040
## model4.7   4 959.8947
## model4.9   4 957.9065
## model4.10  4 959.3308
## model4.11  4 959.9000
```

```
BIC(model4.1,model4.2,model4.3,model4.6,model4.7,model4.9,model4.10,model4.11)
```

```
##           df      BIC
## model4.1   4 967.1638
## model4.2   4 968.2955
## model4.3   4 969.3364
## model4.6   4 965.3614
## model4.7   4 970.1521
## model4.9   4 968.1639
## model4.10  4 969.5882
## model4.11  4 970.1574
```

```
#fit a 5th variable
model5.1 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia,data=heart
)#smallest AIC
model5.2 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+high_blood_pressure+creatinine_phospho
kinase,data=heart)
model5.3 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+diabetes,data=hea
rt)
model5.7 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+high_blood_pressure+platelets,data=hea
rt)
model5.9 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+serum_sodium,data
=heart)
model5.10 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+sex,data=heart)
model5.11 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+smoking,data=hea
rt)
AIC(model5.1,model5.2,model5.3,model5.7,model5.9,model5.10,model5.11)
```

```
##           df      AIC
## model5.1   5 954.5804
## model5.2   5 954.7759
## model5.3   5 956.4253
## model5.7   5 957.0417
## model5.9   5 954.6594
## model5.10  5 956.7960
## model5.11  5 957.1018
```

```
BIC(model5.1,model5.2,model5.3,model5.7,model5.9,model5.10,model5.11)
```

```
##           df      BIC
## model5.1   5 967.4021
## model5.2   5 967.5976
## model5.3   5 969.2471
## model5.7   5 969.8634
## model5.9   5 967.4812
## model5.10  5 969.6177
## model5.11  5 969.9235

#fit a 6th variable
model6.2 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+creatinine
_phosphokinase,data=heart)#smallest AIC
model6.3 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+diabetes,
data=heart)
model6.7 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+platelets,
data=heart)
model6.9 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+serum_sod
ium,data=heart)
model6.10 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+sex,data
=heart)
model6.11 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+smoking,
data=heart)

BIC(model6.2,model6.3,model6.7,model6.9,model6.10,model6.11)

##           df      BIC
## model6.2   6 968.7788
## model6.3   6 971.3334
## model6.7   6 971.8846
## model6.9   6 968.8292
## model6.10  6 971.6326
## model6.11  6 971.9638

#fit a 7th variable
model7.3 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+creatinin
e_phosphokinase+diabetes,data=heart)
model7.7 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+creatinine
_phosphokinase+platelets,data=heart)
model7.9 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+creatinin
e_phosphokinase+serum_sodium,data=heart)##
model7.10 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+creatini
ne_phosphokinase+sex,data=heart)
model7.11 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+creatini
ne_phosphokinase+smoking,data=heart)
AIC(model7.3,model7.7,model7.9,model7.10,model7.11)

##           df      AIC
## model7.3   7 954.6801
## model7.7   7 955.3065
## model7.9   7 951.8277
## model7.10  7 954.8852
## model7.11  7 955.3927

BIC(model7.3,model7.7,model7.9,model7.10,model7.11)

##           df      BIC
## model7.3   7 972.6305
## model7.7   7 973.2569
## model7.9   7 969.7781
## model7.10  7 972.8357
## model7.11  7 973.3431

#fit a 8th variable
model8.3 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+creatinin
e_phosphokinase+serum_sodium+diabetes,data=heart)
model8.7 = coxph(Surv(heart.time,heart.cns) ~age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+creatinine
_phosphokinase+serum_sodium+platelets,data=heart)
model8.10 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+creatini
ne_phosphokinase+serum_sodium+sex,data=heart)
model8.11 = coxph(Surv(heart.time,heart.cns) ~ age+ejection_fraction+serum_creatinine+high_blood_pressure+anaemia+creatini
ne_phosphokinase+serum_sodium+smoking,data=heart)
AIC(model8.3,model8.7,model8.10,model8.11)

##           df      AIC
## model8.3   8 953.4170
## model8.7   8 953.7961
```

```
## model8.10  8 953.1615
## model8.11  8 953.8241

#Interaction term
fit.across1<-  coxph(Surv(heart.time,heart.cns) ~ age*serum_creatinine,data=heart)
fit.across2<-  coxph(Surv(heart.time,heart.cns) ~ ejection_fraction*serum_creatinine,data=heart)
fit.across3<-coxph(Surv(heart.time,heart.cns) ~ serum_creatinine*high_blood_pressure,data=heart)
fit.across4<-coxph(Surv(heart.time,heart.cns) ~ anaemia*high_blood_pressure,data=heart)
AIC(fit.across1,fit.across2,fit.across3,fit.across4)

##             df       AIC
## fit.across1  3  982.5491
## fit.across2  3  981.4386
## fit.across3  3 1001.6256
## fit.across4  3 1017.5627

model1 = coxph(Surv(heart.time,heart.cns)~ age +
                       ejection_fraction +
                       high_blood_pressure +
                       serum_creatinine,data=heart)
exp(confint(model1,level=0.95))

##                         2.5 %    97.5 %
## age                 1.0268342 1.0638187
## ejection_fraction   0.9332112 0.9703973
## high_blood_pressure 1.0585298 2.4244175
## serum_creatinine    1.2414274 1.6124251

model2 = coxph(Surv(heart.time,heart.cns)~ age + anaemia +
                       high_blood_pressure + serum_creatinine,data=heart)
exp(confint(model2,level=0.95))

##                         2.5 %   97.5 %
## age                 1.0249614 1.060812
## anaemia             0.8257476 1.870563
## high_blood_pressure 0.9537637 2.192769
## serum_creatinine    1.1750402 1.470752

#Stratification
model_strata = coxph(Surv(heart.time,heart.cns)~ age +
                       strata(ejection_fraction) +
                       high_blood_pressure +
                       serum_creatinine,data=heart)
model_strata

## Call:
## coxph(formula = Surv(heart.time, heart.cns) ~ age + strata(ejection_fraction) +
##     high_blood_pressure + serum_creatinine, data = heart)
##
##                         coef exp(coef) se(coef)     z        p
## age                 0.046824  1.047938 0.009816 4.770 1.84e-06
## high_blood_pressure 0.655838  1.926757 0.229162 2.862 0.004211
## serum_creatinine    0.339955  1.404884 0.102761 3.308 0.000939
##
## Likelihood ratio test=44.17  on 3 df, p=1.391e-09
## n= 299, number of events= 96
```

setting up machine learning part

```
#setwd('C:/Users/Julia/Box/pstat175/project')
heart = read.csv("heart_failure_clinical_records_dataset.csv",
                 header = TRUE)

heart.ml <- heart
heart.ml <- heart %>%
  mutate(y = factor(heart$DEATH_EVENT, levels=c(0,1),
                    labels=c("censored", "event"))) %>%
  mutate_at(.vars=vars(-y), .funs=scale)
  # scale others

calc_error_rate <- function(predicted.value, true.value){
  return(mean(true.value!=predicted.value))
  }

records = matrix(NA, nrow=2, ncol=1)
colnames(records) <- c("Accuracy")
rownames(records) <- c("knn","random forest")
```

```r
# set seed
set.seed(2020)

# Get rows for training set
train_rows <- sample(x = 1:nrow(heart.ml),
                     size = 224,
                     replace = FALSE)

heart_train <- heart.ml[train_rows, ]
heart_test <- heart.ml[-train_rows, ]

# Check dimensions
dim(heart_train)
```

```
## [1] 224  14
```

```r
dim(heart_test)
```

```
## [1] 75 14
```

```r
nfold = 10
set.seed(1)
folds = seq.int(nrow(heart_train)) %>%   ## sequential obs ids
  cut(breaks = nfold, labels=FALSE) %>% ## sequential fold ids
  sample ## random fold ids

# do.chunk() for k-fold Cross-validation
do.chunk <- function(chunkid, folddef, Xdat, Ydat, k){

  train = (folddef!=chunkid)

  Xtr = Xdat[train,]
  Ytr = Ydat[train]
  Xvl = Xdat[!train,]
  Yvl = Ydat[!train]

  predYtr = knn(train = Xtr, test = Xtr, cl = Ytr, k = k)
  predYvl = knn(train = Xtr, test = Xvl, cl = Ytr, k = k)

  data.frame(train.error = calc_error_rate(predYtr, Ytr),
             val.error=calc_error_rate(predYvl, Yvl))
}

kvec <- c(1, seq(10, 50, length.out=5))

# Set up covariates
Xdat <- heart_train %>%
  dplyr::select(age, anaemia, high_blood_pressure,
                serum_creatinine, time) %>%
  as.matrix()

# Outcome variable
Ydat <- heart_train$y

plan(multiprocess) #parallel fitting

set.seed(1)
out <- expand.grid(k = kvec, chunkid = 1:nfold) %>%
  as_tibble() %>%
  mutate(fit = future_map2(chunkid, k, do.chunk, folddef = folds, Xdat = Xdat, Ydat = Ydat)) %>%
  unnest()

## Warning: `cols` is now required.
## Please use `cols = c(fit)`
```

model 1

```r
# KNN
best.kfold <- 10
Xtest <- heart_test %>%
  dplyr::select(age, anaemia, high_blood_pressure,
                serum_creatinine, time) %>%
  as.matrix()

# Traning data outcome
Ytest <- heart_test$y
```

```r
set.seed(1)
## get classifications for training set
predYtr <- knn(train = Xdat, test = Xdat, cl = Ydat, k = best.kfold)

## get classifications for testing set
predYvl <- knn(train = Xdat, test = Xtest, cl = Ydat, k = best.kfold)

#records[1, 1] <- calc_error_rate(predYtr, Ydat)
records[1, 1] <- (1-calc_error_rate(predYvl, Ytest))
```

Random Forest

```r
# Random Forest
heart.rf <- randomForest( y ~  age + anaemia + high_blood_pressure
                          + serum_creatinine + time,
                    data = heart_train,importance=TRUE)
rf.predict<-predict(heart.rf,newdata=heart_test)

## random forest confusion matrix
rf.err = table(pred = rf.predict, truth = heart_test$y)
test.rf.err = 1 - sum(diag(rf.err))/sum(rf.err)
records[2, 1]=1- test.rf.err

records

##                 Accuracy
## knn           0.7600000
## random forest 0.7733333
```

model 2

```r
# KNN
best.kfold <- 10
Xtest <- heart_test %>%
  dplyr::select(age, ejection_fraction, high_blood_pressure,
                serum_creatinine, time) %>%
  as.matrix()

# Traning data outcome
Ytest <- heart_test$y

set.seed(1)
## get classifications for training set
predYtr <- knn(train = Xdat, test = Xdat, cl = Ydat, k = best.kfold)

## get classifications for testing set
predYvl <- knn(train = Xdat, test = Xtest, cl = Ydat, k = best.kfold)

#records[1, 1] <- calc_error_rate(predYtr, Ydat)
records[1, 1] <- (1-calc_error_rate(predYvl, Ytest))
```

Random Forest

```r
# Random Forest
heart.rf <- randomForest( y ~  age + ejection_fraction + high_blood_pressure
                          + serum_creatinine + time,
                    data = heart_train,importance=TRUE)
rf.predict<-predict(heart.rf,newdata=heart_test)

## random forest confusion matrix
rf.err = table(pred = rf.predict, truth = heart_test$y)
test.rf.err = 1 - sum(diag(rf.err))/sum(rf.err)
records[2, 1]=1- test.rf.err

records

##                 Accuracy
## knn           0.8266667
## random forest 0.8133333
```