Survival Analysis Report

Heart Attack Patients Survival Analysis

Author: Fardil Bhugaloo

Date: 04/05/2022

Abstract

This analysis performs survival analysis to predict survival month ("survival") based on the covariates in Echocardiogram. The survival analysis includes use of censoring data, Kaplan-Meier estimates, Log-rank test, and Cox proportional hazards model. There is little correlation between survival time and the covariates, which makes it hard to derive significant results. However, by exploring Kaplan-Meier estimates, it seems to have difference in survival time in the first two years after heart attacks.

Introduction

Myocardial infarctions, more commonly known as heart attacks, is a serious health concern and are a leading cause of death among adults worldwide and contributes to economic losses and strain on the health care system. According to The American Heart Association Report (Heart and Stroke Statistics - 2022 Update), cardiac arrest remains a public health crisis. There are more than 356,000 out-of-hospital cardiac arrests (OHCA) annually in the U.S., nearly 90% of them fatal. In many instances, heart attacks are preventable, especially with careful monitoring. Methods to predict the survival rate of the patient is key in allocating time and resources to those who need it most.

Survival analysis applies to datasets where we measure events occurring over time and gathers a set of methods to answer some questions such as:

- How much time does it take for an event to occur?
- What is the probability for a patient to survive a certain amount of time given a condition?
- Are there statistically significant differences in the survival time between diverse groups?

Here we used the Echocardiogram data set of patients that suffered heart attacks at some point in the past to perform survival analysis to predict survival months based on the covariates in Echocardiogram.

Data Description

The data used are the echocardiogram dataset available from UCI.[1] It contains the data of 132 patients that suffered heart attacks with 13 attributes.

Attribute Information (From data key):

1. survival -- the number of months patient survived (has survived, if patient is still alive). Because all the patients had their heart attacks at different times, it is possible that some patients have

[1] https://archive.ics.uci.edu/ml/datasets/echocardiogram

survived less than one year but they are still alive. Check the second variable to confirm this. Such patients cannot be used for the prediction task mentioned above.

2. still-alive -- a binary variable. 0=dead at end of survival period, 1 means still alive
3. age-at-heart-attack -- age in years when heart attack occurred
4. pericardial-effusion -- binary. Pericardial effusion is fluid around the heart. 0=no fluid, 1=fluid
5. fractional-shortening -- a measure of contracility around the heart lower numbers are increasingly abnormal
6. epss -- E-point septal separation, another measure of contractility. Larger numbers are increasingly abnormal.
7. lvdd -- left ventricular end-diastolic dimension. This is a measure of the size of the heart at end-diastole. Large hearts tend to be sick hearts.
8. wall-motion-score -- a measure of how the segments of the left ventricle are moving
9. wall-motion-index -- equals wall-motion-score divided by number of segments seen. Usually 12-13 segments are seen in an echocardiogram. Use this variable INSTEAD of the wall motion score.
10. mult -- a derivate var which can be ignored
11. name -- the name of the patient (I have replaced them with "name")
12. group -- meaningless, ignore it
13. alive-at-1 -- Boolean-valued. Derived from the first two attributes. 0 means patient was either dead after 1 year or had been followed for less than 1 year. 1 means patient was alive at 1 year

	survival	alive	age	pericardialeffusion	fractionalshortening	epss	lvdd	wallmotion-score	wallmotion-index	mult	aliveat1
count	130.000000	131.000000	126.000000	132.000000	124.000000	117.000000	121.000000	128.000000	130.00000	129.000000	75.000000
mean	22.182923	0.328244	62.813722	0.765152	0.216734	12.164769	4.763157	14.438125	1.37800	0.786202	0.346667
std	15.858267	0.471377	8.342110	6.697225	0.107513	7.370159	0.810013	5.018566	0.45185	0.225661	0.506534
min	0.030000	0.000000	35.000000	0.000000	0.010000	0.000000	2.320000	2.000000	1.00000	0.140000	0.000000
25%	7.875000	0.000000	57.000000	0.000000	0.150000	7.000000	4.230000	11.000000	1.00000	0.714000	0.000000
50%	23.500000	0.000000	62.000000	0.000000	0.205000	11.000000	4.650000	14.000000	1.21600	0.786000	0.000000
75%	33.000000	1.000000	67.750000	0.000000	0.270000	16.100000	5.300000	16.500000	1.50750	0.857000	1.000000
max	57.000000	1.000000	86.000000	77.000000	0.610000	40.000000	6.780000	39.000000	3.00000	2.000000	2.000000

There are missing values across variables in the dataset, but due to the low number of observations in the dataset, I will only drop missing values for the *survival* and *alive* attributes, and I will impute the other remaining with the mean of each column. After dealing with missing values, we have 130 observations and 8 variables.

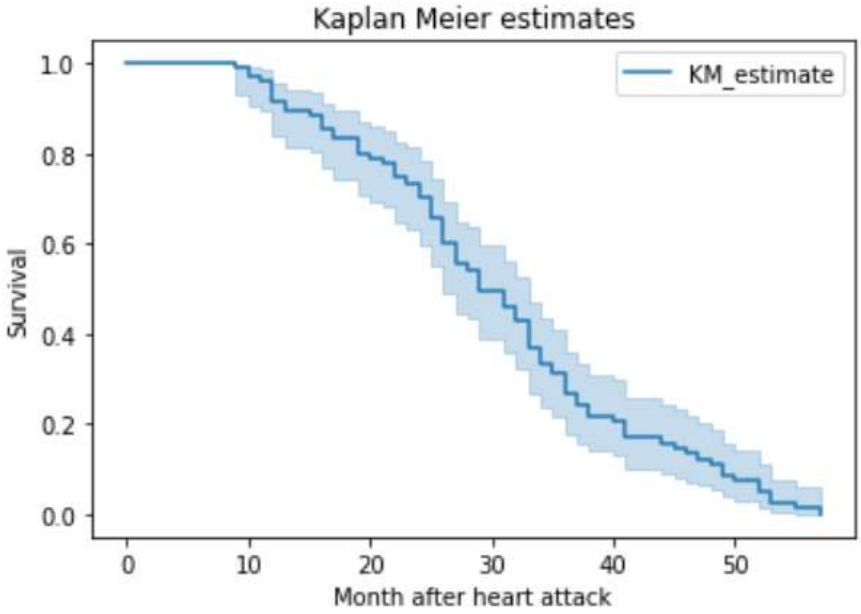
For alive = 1 patients, because they are alive during data collection period and we do not know their survival months after the data collection, they are regarded as censored data. Hence, the data were transformed into censored and non-censored and were saved in a new variable “dead”. After transformation, we have 88 non-censored data and 42 censored data.

Kaplan-Meier Model

The Kaplan-Meier (KM) statistic measures the probability that a patient will survive past a specific point in time. At t=0, the statistic is 1 (100%). When t increases infinitely, the statistic becomes 0, as seen in the equation below.

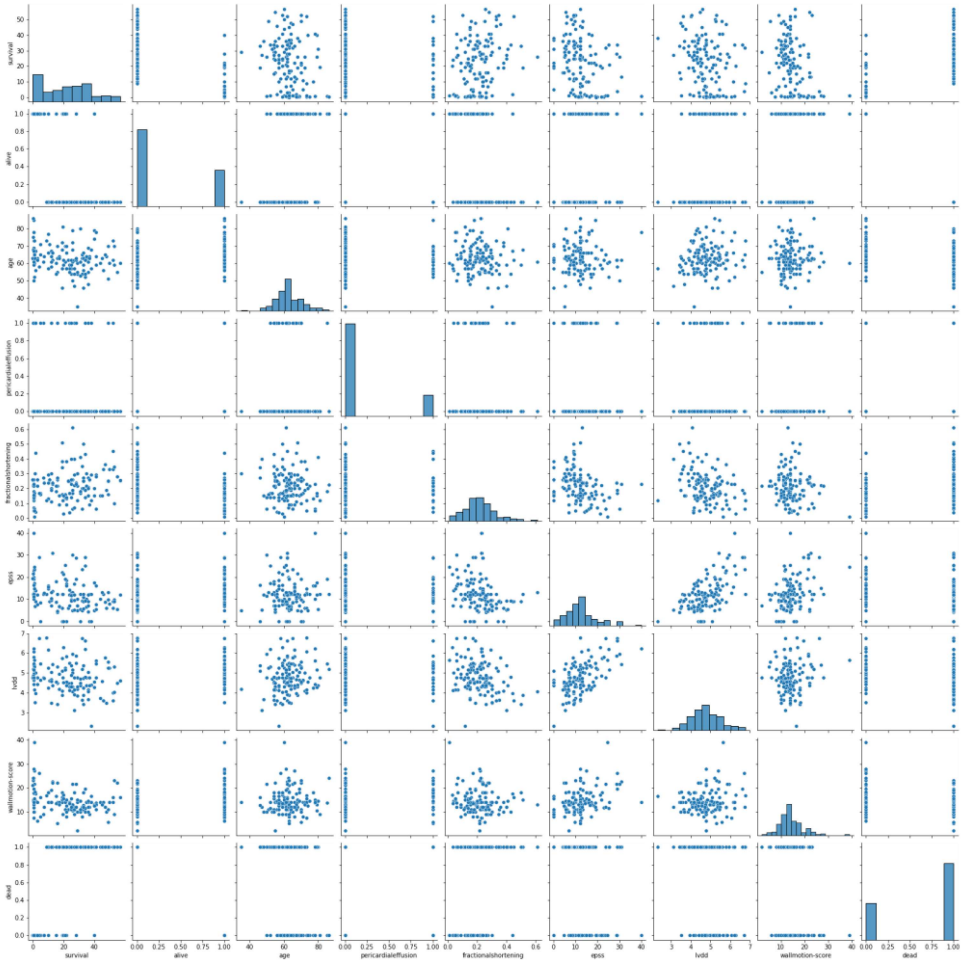
$$\widehat{S}(t) = \prod_{i:\ t_i \leq t} \left(1 - \frac{d_i}{n_i}\right),$$

The plot of the KM estimator is a series of decreasing horizontal steps, approaching the true survival function based on conditional probabilities; each new proportion is conditional on the previous proportion.

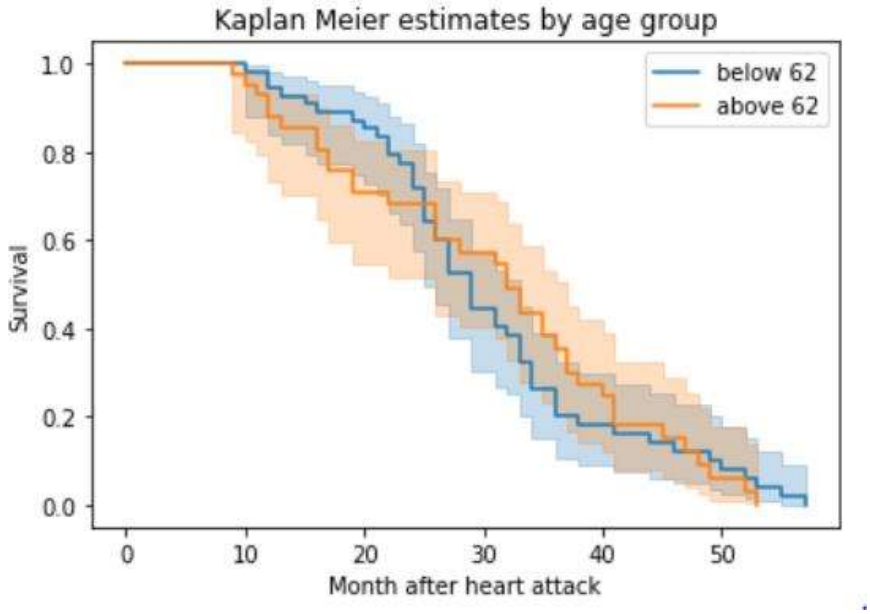


From the KM plot above we can see that the median survival time in month is approximately 28 months

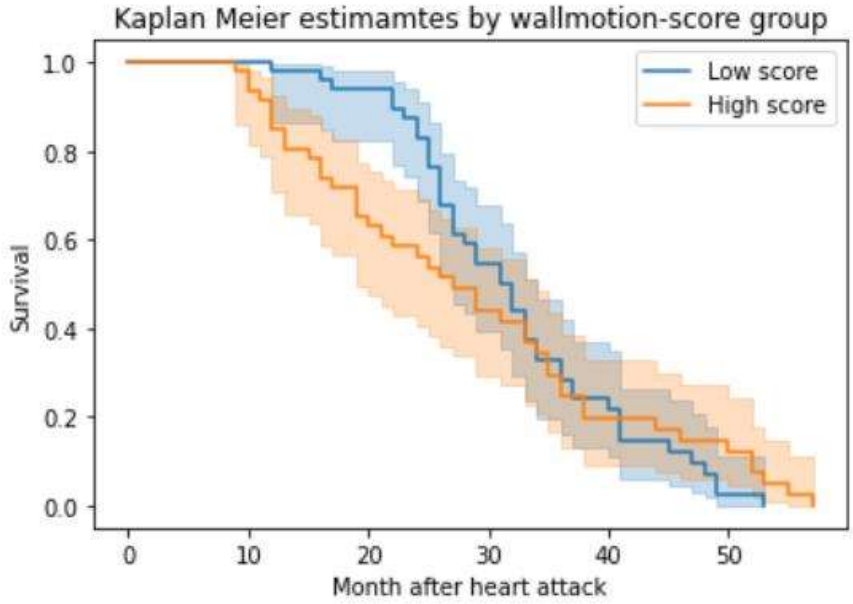
From the correlation plot below, we can see a slightly negative relationship of age and wallmotion-score to survival, so I used median to make two groups within each variable to see difference in survival time.



The median age is 62, so I divided the data into 2 groups; below and above 62. Below is the KM plot for the age group



The median Wallmotion-score is 14, so I grouped the data in low score (< 14) and high score. Below is the KM plot for the wallmotion-score group.



The difference by age groups seems to be weak. However, there seems to differ by wallmotion-score group for the first 24 months (2 years) after heart attack. To test, this hypothesis, I computed the long rank test based on wallmotion-score group.

Log-rank Test

```
<lifelines.StatisticalResult>
               t_0 = -1
 null_distribution = chi squared
degrees_of_freedom = 1

---
test_statistic      p  -log2(p)
          9.98 <0.005      9.31
```

"test_statistic" here is a chi-square statistic. It shows chi-square statistic 9.98, and p-value is less than 5%. Thus, confirming that there is a significant difference in survival time by wallmotion score group for the first 2 year after heart attack.

Cox-Proportional Hazard Model

The Cox model is used to describe the simultaneous effects of several variables on the rate of particular event happening at a specific point in time. Here, I will use the model to study the simultaneous effects of wallmotion score and age on the survival of the patient. This model is defined with the Hazard Function, h(t), which is the hazard that an event happens at a time = t, given than it has not occurred yet, prior to the time t.

model	lifelines.CoxPHFitter
duration col	'survival'
event col	'censored'
baseline estimation	breslow
number of observations	130
number of events observed	28
partial log-likelihood	-117.36
time fit was run	2022-05-03 11:32:28 UTC

	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(coef) upper 95%	cmp to	z	p	-log2(p)
age	0.02	1.02	0.02	-0.03	0.07	0.97	1.07	0.00	0.88	0.38	1.39
wallmotion-score	-1.27	0.28	0.42	-2.09	-0.45	0.12	0.64	0.00	-3.02	<0.005	8.63

Concordance	0.70
Partial AIC	238.71
log-likelihood ratio test	10.68 on 2 df
-log2(p) of ll-ratio test	7.71

```
# p-value of Log-likelihood ratio test
round(stats.chi2.sf(10.68, 2),4)
```

```
0.0048
```

Results and Conclusion

Whether this model is significant or not depends on the result of Log-likelihood ratio test at the bottom of the summary. This statistic follows Chi-square distribution with 2 degree of freedom, and p-value is 0.0048. It says this cox model is significant so that statistical inference is based on this model. Wallmotion-score group is a risk factor for survival time, but age is not by checking p-values. Negative sign of wallmotion-score variable indicates that the patients with low wallmotion score reduce the risk of death. Hazard ration of wallmotion-score is 0.28, which means it reduce in hazard since it is less than 1 and it reduces the hazard by 72% (1 - 0.28). Thus, it can be concluded that for the first two

years after each patient experiences heart attack, the people with high wallmotion score would have a higher risk of death so that we can pay attention to this group of patients.

Appendix

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statistics
from sklearn.impute import SimpleImputer
from lifelines import KaplanMeierFitter, CoxPHFitter
from lifelines.statistics import logrank_test
from scipy import stats
```

In [ ]:
```
# Load the data
data = pd.read_csv("echocardiogram.csv")
# data.head()
```

In [ ]:
```
# Imputing missing data
imp_mean = SimpleImputer(missing_values = np.nan, strategy = 'mean')
COLUMNS = ['age', 'pericardialeffusion', 'fractionalshortening', 'epss',
X = imp_mean.fit_transform(data[COLUMNS])
data_X = pd.DataFrame(X,
                      columns = COLUMNS)
data_X.shape
```

In [ ]:
```
col_keep = ['survival', 'alive']
df_keep = data[col_keep]
df_keep.shape
```

In [ ]:
```
df = pd.concat([df_keep, data_X], axis = 1)
df = df.dropna()
# print(df.isnull().sum())
# print(df.shape)
```

In [ ]:
```
df.loc[df.alive == 1, 'dead'] = 0
df.loc[df.alive == 0, 'dead'] = 1
# df.groupby('dead').count()
```

Kaplan Meier

In [ ]:
```
kmf = KaplanMeierFitter()
T = df['survival']
E = df['dead']
kmf.fit(T, event_observed = E)
kmf.plot()
plt.title("Kaplan Meier estimates")
plt.xlabel("Month after heart attack")
plt.ylabel("Survival")
plt.show()
```

In [ ]:
```
print(statistics.median(df['age']))
print(statistics.median(df['wallmotion-score']))
```

In [ ]:
```
age_group = df['age'] < statistics.median(df['age'])
ax = plt.subplot(111)
kmf.fit(T[age_group], event_observed = E[age_group], label = 'below 62')
kmf.plot(ax = ax)
kmf.fit(T[~age_group], event_observed = E[~age_group], label = 'above 62
kmf.plot(ax = ax)
plt.title("Kaplan Meier estimates by age group")
plt.xlabel("Month after heart attack")
plt.ylabel("Survival")
```

In [ ]:
```
score_group = df['wallmotion-score'] < statistics.median(df['wallmotion-
ax = plt.subplot(111)
kmf.fit(T[score_group], event_observed = E[score_group], label = 'Low sc
kmf.plot(ax = ax)
kmf.fit(T[~score_group], event_observed = E[~score_group], label = 'High
kmf.plot(ax = ax)
plt.title("Kaplan Meier estimamtes by wallmotion-score group")
plt.xlabel("Month after heart attack")
plt.ylabel("Survival")
```

Log-rank test

In [ ]:
```
month_cut = 24
df.loc[(df.dead == 1) & (df.survival <= month_cut), 'censored'] = 1
df.loc[(df.dead == 1) & (df.survival > month_cut), 'censored'] = 0
df.loc[df.dead == 0, 'censored'] = 0
E_v2 = df['censored']

T_low = T[score_group]
T_high = T[~score_group]
E_low = E_v2[score_group]
E_high = E_v2[~score_group]

results = logrank_test(T_low, T_high, event_observed_A = E_low, event_ob
results.print_summary()
```

Cox proportional hazards model

In [ ]:
```
cph = CoxPHFitter()
df_score_group = pd.DataFrame(score_group)
df_model = df[['survival', 'censored', 'age']]
df_model = pd.concat([df_model, df_score_group], axis = 1)
cph.fit(df_model, 'survival', 'censored')
cph.print_summary()
```

In [ ]:
```
# p-value of Log-likelihood ratio test
round(stats.chi2.sf(10.68, 2),4)
```