Heart Failure Prediction

SHRESTHA, SANDESH AND SEO, JUN SOO

Table of Contents

Introduction

This project is about cardiovascular diseases. According to World Health Organization, Cardiovascular diseases are the number 1 cause of death globally, taking an estimated 17.9 million lives in 2019, which accounts for 31% of all deaths worldwide. Of these deaths, 85% were due to heart attack and stroke. In this study we are trying find what factor causes the cardiovascular disease led to heart failure and what features cardiovascular disease penitent to avoid from getting heart failure or prevented cardiovascular disease.

Hence, we wanted to find out the factors that risk a person to heart failure and how long a person can survive if they have specific disease.

About the Data

In this project, we have 12 variables: anemia, creatinine phosphokinase, diabetes, ejection fraction, high blood pressure, platelets, serum creatinine, serum sodium, sex, smoking, time, and death event. We have a total of 6 categorical variables anemia indicates if the observed patient has anemia, 0 does not have anemia, and one indicates the patient has anemia and diabetes, high blood pressure, sex, smoking, death event, and these variables are all binary. We have a total of 6 continuous or discrete variables, which are age, creatinine phosphokinase, ejection fractions, platelets, serum creatinine, serum sodium, and time. Age indicated the patient's age. Anemia is a condition of lack of healthy red blood cells to carry adequate oxygen to a patient's body, according to Mayo Clinic. Creatinine phosphokinase is an enzyme in the body, according to a website called Mount Sinai. Diabetes is the status of if the patient has diabetes or not. Ejection fraction is, according to a website called hart it is the measurement, expressed as a percentage, of how much blood the left ventricle pumps out with each contraction. High blood pressure is indicated if patients have high blood pressure or not. Platelets are the type of blood cell it indicates how much a patient has. According to Mayo Clinic, serum creatinine is the amount of creatinine in inpatient blood. Serum sodium is the measurement of the amount of serum sodium in the patient body. Sex is indicated by the gender of the patient, male or female. Smoking is if the patient is smoking or not. Time is indicated how long it took the patient to die or leave the observation. Beath event is an event of patients dying.

Materials and Methods

We used python for our analysis purpose. Th libraries we used are:

- NumPy
- Pandas
- Lifelines: KaplanMeierFitter, median_survival_times, CoxPHFitter, proportional_hazard_test, WeibullFitter, ExponentialFitter, LogNormalFitter, LogLogisticFitter, LogNormalAFTFitter
- Matplotlib

The methods we approached this dataset with:

- Non-parametric Method
  - Kaplan -Maier
- Semi- Parametric method
  - Cox-Proportional Hazard Model
- Distribution
  - Weibull Distribution
  - Log-Normal Distribution
  - Log-Logistic Distribution
  - Exponential Distribution

Results, Graphs and Tables

- We had a dataset consisted of 299 rows and 12 columns.
- we checked the data types and filtered the null values if found.

```
In [8]: df.dtypes
```

```
Out[8]: anaemia                       int64
        creatinine_phosphokinase      int64
        diabetes                      int64
        ejection_fraction             int64
        high_blood_pressure           int64
        platelets                   float64
        serum_creatinine            float64
        serum_sodium                  int64
        sex                           int64
        smoking                       int64
        time                          int64
        DEATH_EVENT                   int64
        dtype: object
```

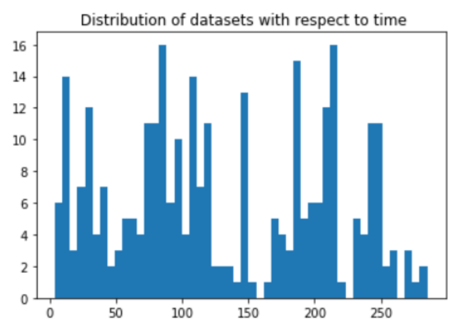
```
In [10]: df.isnull().sum()
```

```
Out[10]: anaemia                     0
         creatinine_phosphokinase    0
         diabetes                    0
         ejection_fraction           0
         high_blood_pressure         0
         platelets                   0
         serum_creatinine            0
         serum_sodium                0
         sex                         0
         smoking                     0
         time                        0
         DEATH_EVENT                 0
         dtype: int64
```

- We plotted the histogram to see the distribution of data with respect to time.

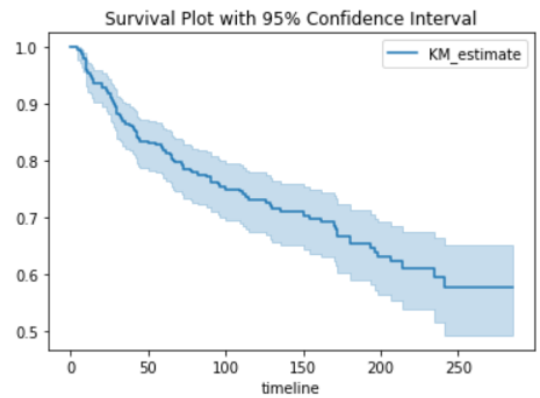
```
In [12]: # Plotting a histogram in order to see the distribution of data with repect of time
```

```
In [19]: t = df['time']
         plt.hist(t, bins=50)
         plt.title(" Distribution of datasets with respect to time")
         plt.show()
```



- As an initial approach we analyze our dataset with non-parametric method called Kaplan Maier. We plotted and fitted the survival probability with two parameters: ‘time’ as duration and ‘DEATH_EVENT’ as event observed with 95% Confidence Interval.

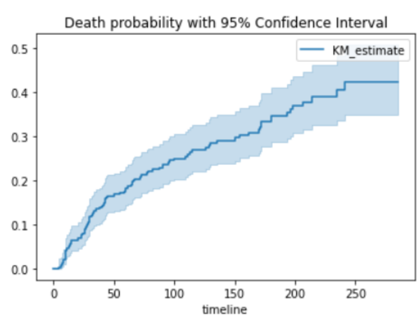
```
In [24]: s = df["DEATH_EVENT"]
         kmf= KaplanMeierFitter()
         kmf.fit(durations = t, event_observed= s)
         kmf.plot_survival_function()
         plt.title("Survival Plot with 95% Confidence Interval")

Out[24]: Text(0.5, 1.0, 'Survival Plot with 95% Confidence Interval')
```



Note: the above plot shows the survival probability overtime time we can see that the survival probability is decreasing as the time is increasing.

- We also plotted the death probability plot with KM estimate which is upside down of the survival plot.

```
In [28]: kmf.plot_cumulative_density()
         plt.title("Death probability with 95% Confidence Interval")

Out[28]: Text(0.5, 1.0, 'Death probability with 95% Confidence Interval')
```



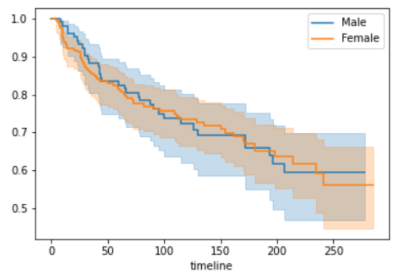
- We calculated the KM median survival time.

```
In [29]: #MEDIAN SURVIVAL TIME

In [32]: md = kmf.median_survival_time_
         md_CI = median_survival_times(kmf.confidence_interval_)
         print(md)
         print(md_CI)

         inf
              KM_estimate_lower_0.95  KM_estimate_upper_0.95
         0.5                   241.0                     inf
```

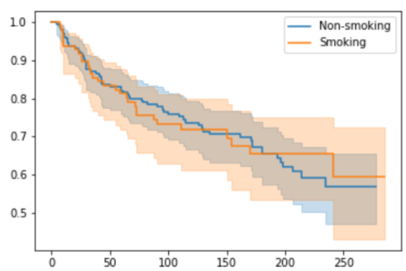
- In order to get the more insights and findings about the datasets we plotted the KM plot for sex and smoking.

```
# KM plot for gender and sex
```

```
ax = plt.subplot(111)
variable = (df["sex"] == 0)
kmf.fit(durations = t[variable], event_observed = s[variable], label="Male")
kmf.plot_survival_function(ax = ax)
kmf.fit(t[~variable], event_observed = s[~variable], label = "Female")
kmf.plot_survival_function(ax = ax, at_risk_counts = True)
plt.title("Survival of different gender group")
```



We saw that male has high probability of survival than female. However, female survival probability is consistent whereas male survival probability seems to be inconsistent.



We saw that smoking has high probability of survival than non-smoking. However, non-smoker survival probability is consistent whereas smoker survival probability seems to be inconsistent over the time.

- Then, we approached semi-parametric Cox-Proportional Hazard

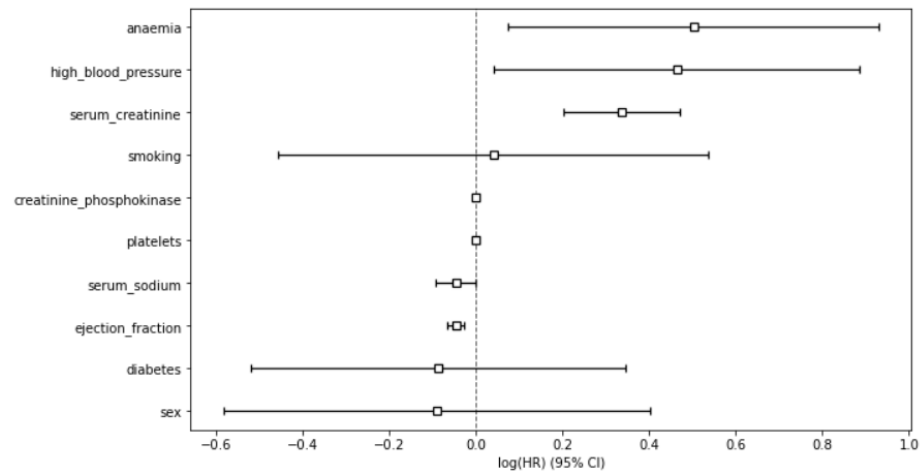
```
In [48]: from lifelines import CoxPHFitter

         cph = CoxPHFitter()
         cph.fit(df, duration_col = 'time', event_col = 'DEATH_EVENT')
         cph.print_summary()
```

time fit was run 2022-05-03 20:57:12 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)
anaemia	0.50	1.65	0.22	0.08	0.93	1.08	2.53	0.00	2.31	0.02	5.57
creatinine_phosphokinase	0.00	1.00	0.00	0.00	0.00	1.00	1.00	0.00	2.06	0.04	4.65
diabetes	-0.09	0.92	0.22	-0.52	0.34	0.60	1.41	0.00	-0.40	0.69	0.53
ejection_fraction	-0.05	0.95	0.01	-0.07	-0.03	0.94	0.97	0.00	-4.52	<0.005	17.27
high_blood_pressure	0.46	1.59	0.22	0.04	0.89	1.04	2.43	0.00	2.16	0.03	5.01
platelets	0.00	1.00	0.00	-0.00	0.00	1.00	1.00	0.00	0.10	0.92	0.11
serum_creatinine	0.34	1.40	0.07	0.20	0.47	1.22	1.60	0.00	4.93	<0.005	20.24
serum_sodium	-0.05	0.96	0.02	-0.09	0.00	0.91	1.00	0.00	-1.94	0.05	4.24
sex	-0.09	0.91	0.25	-0.58	0.40	0.56	1.49	0.00	-0.36	0.72	0.48
smoking	0.04	1.04	0.25	-0.46	0.54	0.63	1.71	0.00	0.16	0.87	0.19

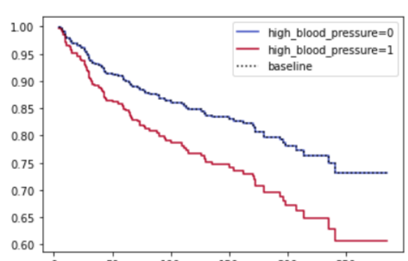
- We plotted the 95% CI of coefficients as well.

```
In [100]: plt.subplots(figsize = (10,6))
          cph.plot()

Out[100]: <matplotlib.axes._subplots.AxesSubplot at 0x22b44cb24f0>
```

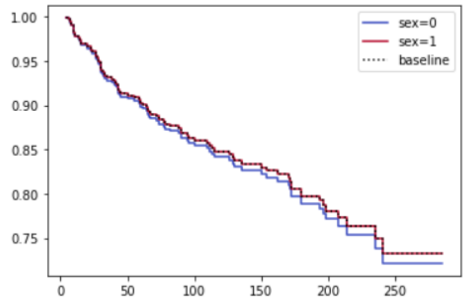


- We plotted the major covariates to see the effects and changes in outcomes.

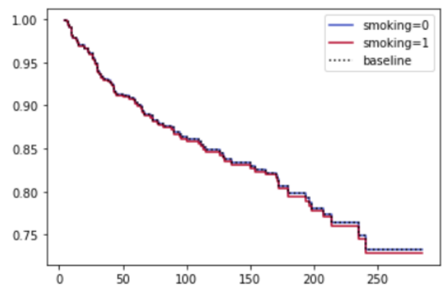
```
In [68]: cph.plot_partial_effects_on_outcome(covariates = 'high_blood_pressure', values=[0,1], cmap= 'coolwarm')

Out[68]: <matplotlib.axes._subplots.AxesSubplot at 0x22b44aa1eb0>
```



In [69]:
```
cph.plot_partial_effects_on_outcome(covariates = 'sex', values=[0,1], cmap= 'coolwarm')
```
Out[69]: <matplotlib.axes._subplots.AxesSubplot at 0x22b44c083a0>



In [70]:
```
cph.plot_partial_effects_on_outcome(covariates = 'smoking', values=[0,1], cmap= 'coolwarm')
```
Out[70]: <matplotlib.axes._subplots.AxesSubplot at 0x22b44c4efa0>



we can see that covariates are making difference in the outcomes for survival probability. We can see major change in survival with the person having high blood pressure. The survival decreased in massive amounts if the person is suffering from high blood pressure, and it makes sense too.

And, for sex and smoking habit. We did not see many changes.

- After seeing the plot we checked the normality assumptions.

```
In [101]: cph.check_assumptions(ab, p_value_threshold = 0.05)
```

```
The ``p_value_threshold`` is set at 0.05. Even under the null hypothesis of no violations, some
covariates will be below the threshold by chance. This is compounded when there are many covariates.
Similarly, when there are lots of observations, even minor deviances from the proportional hazard
assumption will be flagged.

With that in mind, it's best to use a combination of statistical tests and visual tests to determine
the most serious violations. Produce visual plots using ``check_assumptions(..., show_plots=True)``
and looking for non-constant lines. See link [A] below for a full example.
```

null_distribution	chi squared
degrees_of_freedom	1
model	<lifelines.CoxPHFitter: fitted with 299 total ...
test_name	proportional_hazard_test

		test_statistic	p	-log2(p)
anaemia	km	0.03	0.86	0.22
	rank	0.09	0.76	0.39
creatinine_phosphokinase	km	0.64	0.42	1.24
	rank	0.60	0.44	1.19
diabetes	km	0.01	0.92	0.12
	rank	0.00	0.96	0.07
ejection_fraction	km	6.84	0.01	6.81
	rank	7.11	0.01	7.03
high_blood_pressure	km	0.57	0.45	1.15
	rank	0.57	0.45	1.15
platelets	km	0.00	0.95	0.08
	rank	0.06	0.81	0.30
serum_creatinine	km	2.54	0.11	3.18
	rank	2.69	0.10	3.31
serum_sodium	km	1.82	0.18	2.50
	rank	2.57	0.11	3.20
sex	km	0.43	0.51	0.97
	rank	0.74	0.39	1.36

sex	km	0.43	0.51	0.97
	rank	0.74	0.39	1.36
smoking	km	0.73	0.39	1.35
	rank	0.41	0.52	0.94

```
1. Variable 'ejection_fraction' failed the non-proportional test: p-value is 0.0077.

   Advice 1: the functional form of the variable 'ejection_fraction' might be incorrect. That is,
there may be non-linear terms missing. The proportional hazard test used is very sensitive to
incorrect functional forms. See documentation in link [D] below on how to specify a functional form.
```

We can see that ejection_fraction variable is violating the normality assumption at 5% significance level i.e. (0.01 < 0.05). And rest of the variables are following the normality assumptions.

- We did the hazard test as well to be surer and more certain about our above test.

```
In [103]: from lifelines.statistics import proportional_hazard_test
results = proportional_hazard_test(cph, ab, time_transform='rank')
results.print_summary(decimals=3, model="untransformed variables")
```

time_transform	rank
null_distribution	chi squared
degrees_of_freedom	1
model	<lifelines.CoxPHFitter: fitted with 299 total ...
test_name	proportional_hazard_test

	test_statistic	p	-log2(p)
anaemia	0.09	0.76	0.39
creatinine_phosphokinase	0.60	0.44	1.19
diabetes	0.00	0.96	0.07
ejection_fraction	7.11	0.01	7.03
high_blood_pressure	0.57	0.45	1.15
platelets	0.06	0.81	0.30
serum_creatinine	2.69	0.10	3.31
serum_sodium	2.57	0.11	3.20
sex	0.74	0.39	1.36
smoking	0.41	0.52	0.94

We got the same result as above.

- After all these steps, we are ready for our final steps, choosing the best fitted model and distribution.

```
In [113]: from lifelines import WeibullFitter,\
                                 ExponentialFitter,\
                                 LogNormalFitter,\
                                 LogLogisticFitter
# Instantiate each fitter
wb = WeibullFitter()
ex = ExponentialFitter()
log = LogNormalFitter()
loglogis = LogLogisticFitter()

model_li = [wb, ex, log, loglogis]
# Fitin to data
for model in model_li:
    model.fit(durations = ab["time"], event_observed = ab["DEATH_EVENT"])
# Print AIC
    print("The AIC value for", model.__class__.__name__, "is",  model.AIC_)

The AIC value for WeibullFitter is 1344.8757115822702
The AIC value for ExponentialFitter is 1347.081826041116
The AIC value for LogNormalFitter is 1336.54616750603
The AIC value for LogLogisticFitter is 1342.3339662530461
```

We chose Log Normal distribution because it has lowest AIC value among all distributions.

- We fitted the log normal model.

```
In [115]: #Fitting LogNormal Model
```

```
In [126]: from lifelines import LogNormalAFTFitter
          logN_aft = LogNormalAFTFitter()
          logN_aft.fit( ab, duration_col='time', event_col='DEATH_EVENT')
          logN_aft.print_summary()
```

model	lifelines.LogNormalAFTFitter
duration col	'time'
event col	'DEATH_EVENT'
number of observations	299
number of events observed	96
log-likelihood	-641.55
time fit was run	2022-05-04 15:02:09 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)
mu_	**anaemia**	-0.58	0.56	0.26	-1.10	-0.07	0.33	0.94	0.00	-2.21	0.03	5.22
	creatinine_phosphokinase	-0.00	1.00	0.00	-0.00	0.00	1.00	1.00	0.00	-1.69	0.09	3.46
	diabetes	0.04	1.04	0.26	-0.47	0.56	0.62	1.75	0.00	0.17	0.87	0.21
	ejection_fraction	0.04	1.04	0.01	0.02	0.06	1.02	1.06	0.00	3.31	<0.005	10.09
	high_blood_pressure	-0.60	0.55	0.26	-1.12	-0.08	0.33	0.92	0.00	-2.28	0.02	5.46
	platelets	0.00	1.00	0.00	-0.00	0.00	1.00	1.00	0.00	0.39	0.70	0.52
	serum_creatinine	-0.42	0.66	0.11	-0.63	-0.21	0.53	0.81	0.00	-3.84	<0.005	13.01
	serum_sodium	0.07	1.07	0.03	0.01	0.12	1.01	1.13	0.00	2.31	0.02	5.57
	sex	0.03	1.03	0.31	-0.57	0.63	0.57	1.88	0.00	0.10	0.92	0.12
	smoking	-0.08	0.93	0.30	-0.67	0.52	0.51	1.67	0.00	-0.25	0.80	0.32
	Intercept	-3.43	0.03	3.82	-10.92	4.06	0.00	57.83	0.00	-0.90	0.37	1.44
sigma_	**Intercept**	0.54	1.72	0.08	0.38	0.70	1.47	2.02	0.00	6.72	<0.005	35.64

Concordance	0.71
AIC	1307.10
log-likelihood ratio test	49.44 on 10 df
-log2(p) of ll-ratio test	21.50

We can see that creatinine and platelets has zero coefficients, so we dropped them.

```
In [128]: # WE can see the variable "creatinine" and paltelets have 0 cofficient so we can drop them.
```

```
In [132]: df_new =  df.drop('platelets', axis=1)
          df_new1 = df_new.drop('creatinine_phosphokinase', axis=1)
          df_new1.head()
```

Out[132]:

	anaemia	diabetes	ejection_fraction	high_blood_pressure	serum_creatinine	serum_sodium	sex	smoking	time	DEATH_EVENT
age										
75.0	0	0	20	1	1.9	130	1	0	4	1
55.0	0	0	38	0	1.1	136	1	0	6	1
65.0	0	0	20	0	1.3	129	1	1	7	1
50.0	1	0	20	0	1.9	137	1	0	7	1
65.0	1	1	20	0	2.7	116	0	0	8	1

```
In [ ]:
```

The new fitted model was plotted:

```
In [137]:
logN_aft = LogNormalAFTFitter()
logN_aft.fit( dfT, duration_col='time', event_col='DEATH_EVENT')
logN_aft.print_summary()
```

model	lifelines.LogNormalAFTFitter
duration col	'time'
event col	'DEATH_EVENT'
number of observations	47
number of events observed	28
log-likelihood	-149.01
time fit was run	2022-05-04 15:13:17 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)
mu_	anaemia	-1.08	0.34	0.51	-2.08	-0.08	0.13	0.93	0.00	-2.11	0.03	4.85
	diabetes	1.30	3.67	0.50	0.31	2.29	1.36	9.86	0.00	2.57	0.01	6.63
	ejection_fraction	0.04	1.04	0.02	-0.00	0.07	1.00	1.08	0.00	1.83	0.07	3.91
	high_blood_pressure	-0.27	0.77	0.47	-1.18	0.65	0.31	1.91	0.00	-0.57	0.57	0.81
	serum_creatinine	-0.05	0.95	0.13	-0.29	0.20	0.75	1.22	0.00	-0.39	0.70	0.52
	serum_sodium	0.16	1.17	0.05	0.05	0.26	1.06	1.30	0.00	2.97	<0.005	8.41
	sex	-0.62	0.54	0.56	-1.71	0.47	0.18	1.60	0.00	-1.12	0.26	1.93
	smoking	0.31	1.37	0.54	-0.74	1.37	0.48	3.92	0.00	0.58	0.56	0.84
	Intercept	-17.82	0.00	7.23	-31.99	-3.65	0.00	0.03	0.00	-2.47	0.01	6.19
sigma_	Intercept	0.29	1.34	0.14	0.01	0.57	1.01	1.77	0.00	2.02	0.04	4.54

sigma_ Intercept 0.29 1.34 0.14 0.01 0.57 1.01 1.77 0.00 2.02 0.04 4.54

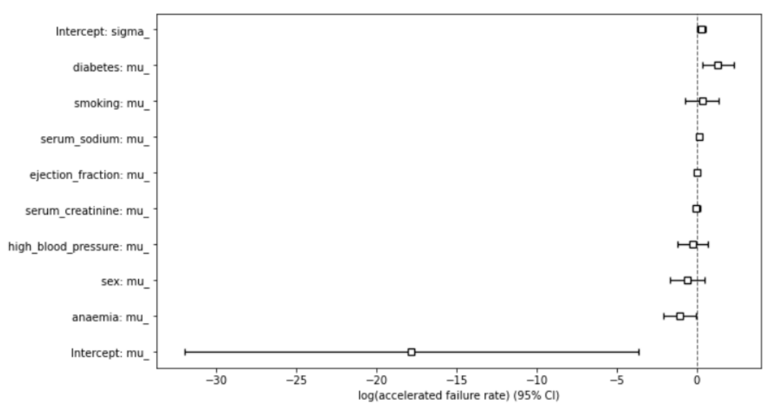
Concordance	0.73
AIC	318.02
log-likelihood ratio test	16.42 on 8 df
-log2(p) of ll-ratio test	4.77

- We estimated the log Normal median survival time.

```
In [138]: #calculating mean median surviaval time

In [141]: print("LogNormalaftFitter median survival time : ",logN_aft.median_survival_time_)
          print("LogNormalaftFitter mean survival time : ",logN_aft.mean_survival_time_)

          LogNormalaftFitter median survival time :  27.187913910199903
          LogNormalaftFitter mean survival time :  66.28391971747246
```

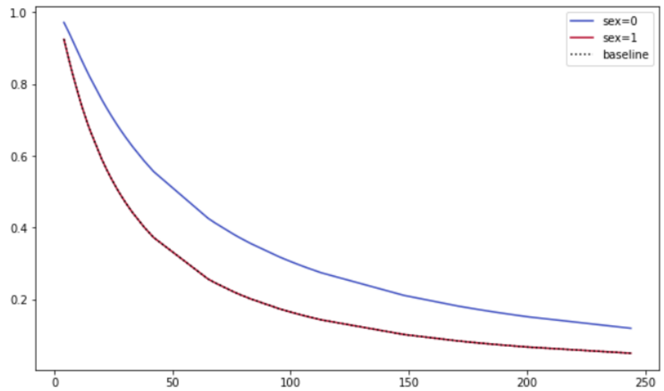
- Plotted the coefficients.

```
Out[144]: <matplotlib.axes._subplots.AxesSubplot at 0x22b44ebc3d0>
```

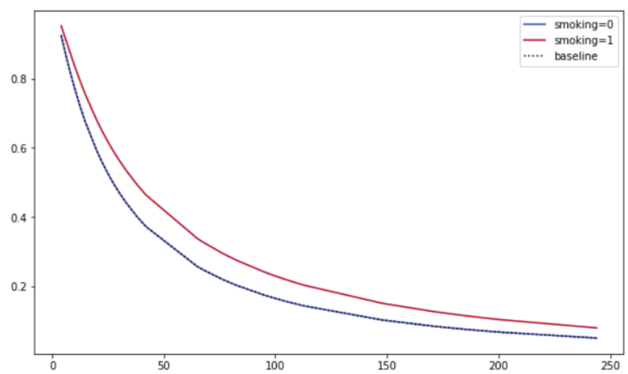


- We plotted the major covariates to see the effects and changes in outcomes.

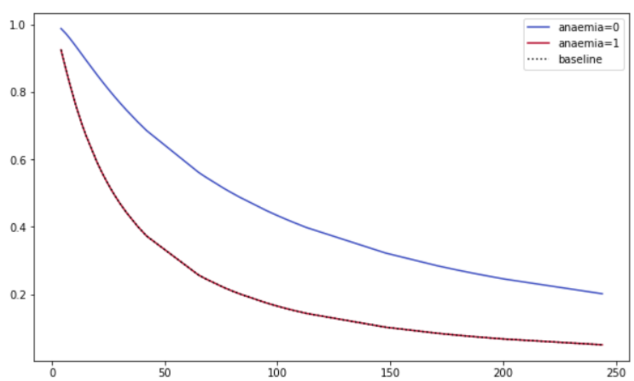
```
In [154]: plt.subplots(figsize=(10, 6))
          logN_aft.plot_partial_effects_on_outcome('sex', values=[0,1], cmap='coolwarm')

Out[154]: <matplotlib.axes._subplots.AxesSubplot at 0x22b44b36e20>
```




```
In [159]: plt.subplots(figsize=(10, 6))
          logN_aft.plot_partial_effects_on_outcome('smoking', values=[0,1], cmap='coolwarm')

Out[159]: <matplotlib.axes._subplots.AxesSubplot at 0x22b46576a60>
```




```
In [161]: plt.subplots(figsize=(10, 6))
          logN_aft.plot_partial_effects_on_outcome('anaemia', values=[0,1], cmap='coolwarm')

Out[161]: <matplotlib.axes._subplots.AxesSubplot at 0x22b4671bd30>
```



We can see that female patients have higher survival probabilities at any given instance of time compared to male patients. Similar, the patient who has anemia has higher probability of dying than the patients without anemia whereas the smokers seem to have higher probabilities of surviving than non-smokers.

Discussion

We started our analysis with non-parametric approach with Kaplan-Maier method. We saw that the survival probability decreased with respect to time where as the death probability increased with respect to time. Then we calculated the median survival time of the KM model, 95% CI interval was found to be (241.0, ∞). We wanted to find out how gender and smoking habit is affecting the model, so plotted the dataset which gave the finding that, male has high probability of survival than female. However, female survival probability is consistent whereas male survival probability seems to be inconsistent. Similarly, smoking has high probability of survival than non-smoking. However, non-smoker survival probability is consistent whereas smoker survival probability seems to be inconsistent.

After that, we analyzed our datasets with semi-parametric Cox-Proportional Hazard methodology. We fitted the model and found out that the creatinine and platelets were not playing major role in the model as there coefficients were zero, so we drop those columns in order to get the good findings and close analysis. After dropping them we plotted the 95% CI of the coefficients, we found out that, except for anemia, high blood pressure and serum_creatinine, all other variables had negative lower intervals.

we wanted more about the datasets, so we plotted the partial effects on survival changes of each categorical co-variates to observe the changes in survival rate. we saw interesting and surprising change in high blood pressure. The person with high blood pressure had low rate of survival i.e (0.62) at 250 days whereas the person with no blood pressure had survival rate of 0.78 at 250 days. However, we did not see much changes in terms of sex and smoking habits.

The last steps for Cox-PH model was to check the normality assumptions:
We found out that, ‘ejection_fraction’ was only variable which was violating the normality assumption.

Note: Normality assumption check is shown above in the graph and tables.

The final step was to select the suitable and best model for our datasets. We know that the model which has lowest AIC value will be considered as the best model. We calculated the AIC value for:

- Weibull Distribution
- Log-Normal Distribution
- Log-Logistic Distribution
- Exponential Distribution

```
The AIC value for WeibullFitter is 1344.8757115822702
The AIC value for ExponentialFitter is 1347.081826041116
The AIC value for LogNormalFitter is 1336.54616750603
The AIC value for LogLogisticFitter is 1342.3339662530461
```

We can see that AIC value for LogNormalFitter is the lowest value. Hence, we chose LogNormal model

We fitted the model and coefficients are:

```
In [137]:
logN_aft = LogNormalAFTFitter()
logN_aft.fit( dfT, duration_col='time', event_col='DEATH_EVENT')
logN_aft.print_summary()
```

model	lifelines.LogNormalAFTFitter
duration col	'time'
event col	'DEATH_EVENT'
number of observations	47
number of events observed	28
log-likelihood	-149.01
time fit was run	2022-05-04 15:13:17 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)
mu_	anaemia	-1.08	0.34	0.51	-2.08	-0.08	0.13	0.93	0.00	-2.11	0.03	4.85
	diabetes	1.30	3.67	0.50	0.31	2.29	1.36	9.86	0.00	2.57	0.01	6.63
	ejection_fraction	0.04	1.04	0.02	-0.00	0.07	1.00	1.08	0.00	1.83	0.07	3.91
	high_blood_pressure	-0.27	0.77	0.47	-1.18	0.65	0.31	1.91	0.00	-0.57	0.57	0.81
	serum_creatinine	-0.05	0.95	0.13	-0.29	0.20	0.75	1.22	0.00	-0.39	0.70	0.52
	serum_sodium	0.16	1.17	0.05	0.05	0.26	1.06	1.30	0.00	2.97	<0.005	8.41
	sex	-0.62	0.54	0.56	-1.71	0.47	0.18	1.60	0.00	-1.12	0.26	1.93
	smoking	0.31	1.37	0.54	-0.74	1.37	0.48	3.92	0.00	0.58	0.56	0.84
	Intercept	-17.82	0.00	7.23	-31.99	-3.65	0.00	0.03	0.00	-2.47	0.01	6.19
sigma_	Intercept	0.29	1.34	0.14	0.01	0.57	1.01	1.77	0.00	2.02	0.04	4.54

We also calculated the mean and median survival time of person in this model:

```
LogNormalaftFitter median survival time :  27.187913910199903
LogNormalaftFitter mean survival time :  66.28391971747246
```

We plotted the covariates graphs to see the changes in survival time. We saw drastic changes in person having anemia and in gender. Whereas, smoking habit was quite surprising for us, the analysis, told that non-smokers has low rate of survival as time increase whereas smoker has higher rate of survival compared to non-smoker.

Conclusion

Therefore, after seeing the outcomes from the data analysis, we found followings:

- Heart failure rate is more in male than female.
- If the person suffers from anemia, the heart failure rate increases drastically and survives less than the person without anemia.
- The smoking habit is not the major cause of heart failure.
- Person with high blood pressure should be considered as highest risk category group of people who might die early due to heart failure.
- The creatinine level and platelets numbers is not the cause for heart failure.
- The median survival time for a person who is male, smoker, has high blood pressure, anemia is 28 and mean is 67.

Citation:

Introduction to survival analysis with scikit-survival¶. survival. (n.d.). Retrieved May 6, 2022, from https://scikit-survival.readthedocs.io/en/stable/user_guide/00-introduction.html

Pandey, A. (2020, April 24). *Survival analysis: Intuition & implementation in python*. Medium. Retrieved May 6, 2022, from https://towardsdatascience.com/survival-analysis-intuition-implementation-in-python-504fde4fcf8e

A complete guide to survival analysis in python, part 1. KDnuggets. (n.d.). Retrieved May 6, 2022, from https://www.kdnuggets.com/2020/07/complete-guide-survival-analysis-python-part1.html