In [127…

```
import pandas as pd
import numpy as np
from lifelines import CoxPHFitter
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.impute import SimpleImputer
from lifelines import AalenAdditiveFitter
from lifelines.datasets import load_rossi

# Load the data from the .xlsx file

# Load the data from the .xlsx file
data = pd.read_excel('data1.xlsx')

# Define categorical variables
categorical_cols = ['SEX', 'CompositeStage', 'LNInvolment', 'Comorbidity', 'FamiliyHistoryOfCancer']
data[categorical_cols] = data[categorical_cols].astype('category')

# Handle missing values in other columns
imputer = SimpleImputer(strategy='median')
data[['DEATH', 'AGE', 'CompositeStage', 'LNInvolment', 'Comorbidity']] = imputer.fit_transform(data[['DEATH', 'AGE', 'Composit

# Standardize the covariates
scaler = StandardScaler()
data[['DEATH', 'AGE', 'CompositeStage', 'LNInvolment', 'Comorbidity']] = scaler.fit_transform(data[['DEATH', 'AGE', 'Composite


# Create a new DataFrame with the required columns for the Buckley-James estimator
buckley_james_data = data[['Months', 'DEATH', 'AGE', 'SEX', 'CompositeStage', 'LNInvolment', 'Comorbidity', 'FamiliyHistoryOfC

# Fit the Buckley-James model with custom options
cph = CoxPHFitter(penalizer=0.1)  # Set the penalizer parameter to control overfitting
cph.fit(buckley_james_data, 'Months', 'DEATH', show_progress=True)  # Set the step_size parameter to control the convergence s

# Print the estimated coefficients (summary)
print(cph.summary)
```

```
Iteration 1: norm_delta = 0.66384, step_size = 0.9500, log_lik = -1663.17959, newton_decrement = 46.04648, seconds_since_start
= 0.0
Iteration 2: norm_delta = 0.03630, step_size = 0.9500, log_lik = -1620.53093, newton_decrement = 0.19362, seconds_since_start =
0.0
Iteration 3: norm_delta = 0.00176, step_size = 0.9500, log_lik = -1620.33817, newton_decrement = 0.00043, seconds_since_start =
0.1
Iteration 4: norm_delta = 0.00000, step_size = 1.0000, log_lik = -1620.33774, newton_decrement = 0.00000, seconds_since_start =
0.1
Convergence success after 4 iterations.
                           coef  exp(coef)  se(coef)  coef lower 95%  \
covariate
AGE                    0.019975   1.020175  0.055896       -0.089580
SEX                    0.027013   1.027381  0.106745       -0.182203
CompositeStage         0.531571   1.701603  0.061434        0.411162
LNInvolment           -0.275748   0.759004  0.053051       -0.379725
Comorbidity           -0.034023   0.966549  0.054884       -0.141594
FamiliyHistoryOfCancer  0.003465   1.003471  0.156806       -0.303870

                        coef upper 95%  exp(coef) lower 95%  \
covariate
AGE                           0.129529             0.914315
SEX                           0.236229             0.833432
CompositeStage                0.651980             1.508570
LNInvolment                  -0.171771             0.684049
Comorbidity                   0.073548             0.867974
FamiliyHistoryOfCancer        0.310800             0.737957

                        exp(coef) upper 95%  cmp to         z             p  \
covariate
AGE                                1.138292     0.0  0.357349  7.208303e-01
SEX                                1.266464     0.0  0.253064  8.002191e-01
CompositeStage                     1.919337     0.0  8.652682  5.030319e-18
LNInvolment                        0.842172     0.0 -5.197833  2.016254e-07
Comorbidity                        1.076320     0.0 -0.619903  5.353217e-01
FamiliyHistoryOfCancer             1.364517     0.0  0.022100  9.823684e-01

                         -log2(p)
covariate
AGE                      0.472268
SEX                      0.321533
```

```
CompositeStage          57.464056
LNInvolment             22.241820
Comorbidity              0.901522
FamiliyHistoryOfCancer   0.025664
```

In [129…

```python
univariate_results = []
for col in data.columns:
    if col not in ['Months', 'ID','DEATH']:
        cph_univariate = CoxPHFitter(penalizer=0.1)
        cph_univariate.fit(data[[col, 'Months', 'DEATH']], 'Months', 'DEATH', show_progress=True)
        univariate_results.append((col, cph_univariate.summary))

# Print the summaries of the univariate analysis
for col, summary in univariate_results:
    print(f"Univariate analysis of: {col}")
    print(summary)
    print("\n")
```

```
Iteration 1: norm_delta = 0.01879, step_size = 0.9500, log_lik = -1663.17959, newton_decrement = 0.06380, seconds_since_start =
0.0
Iteration 2: norm_delta = 0.00085, step_size = 0.9500, log_lik = -1663.11614, newton_decrement = 0.00013, seconds_since_start =
0.0
Iteration 3: norm_delta = 0.00004, step_size = 0.9500, log_lik = -1663.11600, newton_decrement = 0.00000, seconds_since_start =
0.1
Iteration 4: norm_delta = 0.00000, step_size = 1.0000, log_lik = -1663.11600, newton_decrement = 0.00000, seconds_since_start =
0.1
Convergence success after 4 iterations.
Iteration 1: norm_delta = 0.01792, step_size = 0.9500, log_lik = -1663.17959, newton_decrement = 0.06049, seconds_since_start =
0.0
Iteration 2: norm_delta = 0.00095, step_size = 0.9500, log_lik = -1663.11915, newton_decrement = 0.00017, seconds_since_start =
0.0
Iteration 3: norm_delta = 0.00005, step_size = 0.9500, log_lik = -1663.11898, newton_decrement = 0.00000, seconds_since_start =
0.0
Iteration 4: norm_delta = 0.00000, step_size = 1.0000, log_lik = -1663.11898, newton_decrement = 0.00000, seconds_since_start =
0.0
Convergence success after 4 iterations.
Iteration 1: norm_delta = 0.43056, step_size = 0.9500, log_lik = -1663.17959, newton_decrement = 27.12197, seconds_since_start
= 0.0
Iteration 2: norm_delta = 0.04153, step_size = 0.9500, log_lik = -1635.53782, newton_decrement = 0.22899, seconds_since_start =
0.0
Iteration 3: norm_delta = 0.00238, step_size = 0.9500, log_lik = -1635.30845, newton_decrement = 0.00074, seconds_since_start =
0.0
Iteration 4: norm_delta = 0.00000, step_size = 1.0000, log_lik = -1635.30771, newton_decrement = 0.00000, seconds_since_start =
0.0
Convergence success after 4 iterations.
Iteration 1: norm_delta = 0.13600, step_size = 0.9500, log_lik = -1663.17959, newton_decrement = 3.86282, seconds_since_start =
0.0
Iteration 2: norm_delta = 0.01328, step_size = 0.9500, log_lik = -1659.23281, newton_decrement = 0.03364, seconds_since_start =
0.0
Iteration 3: norm_delta = 0.00074, step_size = 0.9500, log_lik = -1659.19915, newton_decrement = 0.00010, seconds_since_start =
0.0
Iteration 4: norm_delta = 0.00000, step_size = 1.0000, log_lik = -1659.19905, newton_decrement = 0.00000, seconds_since_start =
0.0
Convergence success after 4 iterations.
Iteration 1: norm_delta = 0.06577, step_size = 0.9500, log_lik = -1663.17959, newton_decrement = 0.79658, seconds_since_start =
0.0
Iteration 2: norm_delta = 0.00275, step_size = 0.9500, log_lik = -1662.38897, newton_decrement = 0.00141, seconds_since_start =
0.0
```

```
Iteration 3: norm_delta = 0.00014, step_size = 0.9500, log_lik = -1662.38756, newton_decrement = 0.00000, seconds_since_start =
0.0
Iteration 4: norm_delta = 0.00000, step_size = 1.0000, log_lik = -1662.38756, newton_decrement = 0.00000, seconds_since_start =
0.0
Convergence success after 4 iterations.
Iteration 1: norm_delta = 0.02937, step_size = 0.9500, log_lik = -1663.17959, newton_decrement = 0.15086, seconds_since_start =
0.0
Iteration 2: norm_delta = 0.00064, step_size = 0.9500, log_lik = -1663.03168, newton_decrement = 0.00008, seconds_since_start =
0.0
Iteration 3: norm_delta = 0.00003, step_size = 0.9500, log_lik = -1663.03161, newton_decrement = 0.00000, seconds_since_start =
0.0
Iteration 4: norm_delta = 0.00000, step_size = 1.0000, log_lik = -1663.03161, newton_decrement = 0.00000, seconds_since_start =
0.0
Convergence success after 4 iterations.
Univariate analysis of: AGE
               coef  exp(coef)  se(coef)  coef lower 95%  coef upper 95%  \
covariate
AGE       -0.018672   0.981501  0.052274       -0.121127        0.083782

           exp(coef) lower 95%  exp(coef) upper 95%  cmp to         z  \
covariate
AGE                   0.885921             1.087392     0.0 -0.357205

                  p  -log2(p)
covariate
AGE        0.720938  0.472052


Univariate analysis of: SEX
              coef  exp(coef)  se(coef)  coef lower 95%  coef upper 95%  \
covariate
SEX        0.03669   1.037371  0.105493       -0.170073        0.243452

           exp(coef) lower 95%  exp(coef) upper 95%  cmp to         z  \
covariate
SEX                   0.843604             1.275645     0.0  0.347792

                  p  -log2(p)
covariate
SEX        0.727996  0.457997
```

```
Univariate analysis of: CompositeStage
                    coef  exp(coef)  se(coef)  coef lower 95%  coef upper 95%  \
covariate
CompositeStage  0.450207   1.568636   0.06173        0.329217        0.571196

                exp(coef) lower 95%  exp(coef) upper 95%  cmp to        z  \
covariate
CompositeStage              1.38988             1.770383     0.0  7.29312

                           p   -log2(p)
covariate
CompositeStage  3.028591e-13  41.586419


Univariate analysis of: LNInvolment
                coef  exp(coef)  se(coef)  coef lower 95%  coef upper 95%  \
covariate
LNInvolment -0.14234   0.867326  0.051397       -0.243076       -0.041604

             exp(coef) lower 95%  exp(coef) upper 95%  cmp to         z  \
covariate
LNInvolment             0.784212              0.95925     0.0 -2.769423

                   p  -log2(p)
covariate
LNInvolment  0.005616  7.476353


Univariate analysis of: Comorbidity
                 coef  exp(coef)  se(coef)  coef lower 95%  coef upper 95%  \
covariate
Comorbidity -0.065132   0.936944  0.051635       -0.166334         0.03607

             exp(coef) lower 95%  exp(coef) upper 95%  cmp to         z  \
covariate
Comorbidity             0.846764             1.036728     0.0 -1.261403

                   p  -log2(p)
```

```
covariate
Comorbidity  0.207164  2.271156


Univariate analysis of: FamiliyHistoryOfCancer
                             coef  exp(coef)  se(coef)  coef lower 95%  \
covariate
FamiliyHistoryOfCancer  0.085227   1.088964  0.155194       -0.218948

                         coef upper 95%  exp(coef) lower 95%  \
covariate
FamiliyHistoryOfCancer         0.389401             0.803364

                         exp(coef) upper 95%  cmp to         z         p  \
covariate
FamiliyHistoryOfCancer             1.476097     0.0  0.549163  0.582894

                         -log2(p)
covariate
FamiliyHistoryOfCancer  0.778696
```

In [128…

```python
#univariate_results = []
univariate_aic_bic = []
for col in data.columns:
    if col not in ['Months', 'ID']:
        n = len(data)
        llf = cph_univariate.log_likelihood_
        k = cph_univariate.params_.shape[0]
        aic = -2 * llf + 2 * k
        bic = -2 * llf + k * np.log(n)
        univariate_aic_bic.append((col, aic, bic))
        print(f"\nAIC value of {col}:", aic)
        print(f"BIC value of {col}:", bic)
```

```
AIC value of DEATH: 3328.0632186349508
BIC value of DEATH: 3331.9009490821168

AIC value of AGE: 3328.0632186349508
BIC value of AGE: 3331.9009490821168

AIC value of SEX: 3328.0632186349508
BIC value of SEX: 3331.9009490821168

AIC value of CompositeStage: 3328.0632186349508
BIC value of CompositeStage: 3331.9009490821168

AIC value of LNInvolment: 3328.0632186349508
BIC value of LNInvolment: 3331.9009490821168

AIC value of Comorbidity: 3328.0632186349508
BIC value of Comorbidity: 3331.9009490821168

AIC value of FamiliyHistoryOfCancer: 3328.0632186349508
BIC value of FamiliyHistoryOfCancer: 3331.9009490821168
```

In [118…

```python
significant_variables_multivariate = [(var, summary) for var, summary in multivariate_results if summary['p'][var] < 0.05]
print("\nSignificant variables from univariate analysis:")
for var, summary in significant_variables_multivariate:
    print(f"\n{var}:")
    print(summary)
```

```
Significant variables from univariate analysis:

CompositeStage:
                    coef  exp(coef)  se(coef)  coef lower 95%  coef upper 95%  \
covariate
CompositeStage  0.451465   1.570611  0.061942        0.330061        0.572868
AGE             0.013370   1.013460  0.053478       -0.091446        0.118185

                exp(coef) lower 95%  exp(coef) upper 95%  cmp to         z  \
covariate
CompositeStage             1.391053             1.773346     0.0  7.288543
AGE                        0.912611             1.125453     0.0  0.250006

                           p   -log2(p)
covariate
CompositeStage  3.133250e-13  41.537405
AGE             8.025824e-01   0.317279

LNInvolment:
                 coef  exp(coef)  se(coef)  coef lower 95%  coef upper 95%  \
covariate
LNInvolment -0.143911   0.865965  0.051475       -0.244800       -0.043022
AGE         -0.027609   0.972769  0.052098       -0.129719        0.074502

             exp(coef) lower 95%  exp(coef) upper 95%  cmp to         z  \
covariate
LNInvolment             0.782861             0.957891     0.0 -2.795740
AGE                     0.878342             1.077347     0.0 -0.529935

                   p  -log2(p)
covariate
LNInvolment  0.005178  7.593362
AGE          0.596157  0.746236
```

In [119…

```python
# Identify the significant variables from the univariate analysis
significant_variables = [(var, p_value) for var, p_value in univariate_results if p_value < 0.05]

# Convert significant variables to categorical variables
#for var, _ in significant_variables:
  #  data[var] = data[var].astype('category')
```

```python
# One-hot encode the updated categorical variablefor var, _ in significant_variables:
data[var] = data[var].astype('category')

# Print the updated data with significant variables as categorical data
print("Updated data with significant variables as categorical data:")
print(data)

data_encoded = pd.get_dummies(data, columns=[var for var, _ in significant_variables], drop_first=True)

# Update the Buckley-James data with the new categorical variables
buckley_james_data = data_encoded[['Months', 'DEATH', 'AGE'] + [col for col in data_encoded.columns if col.startswith('SEX_')
```

```
Updated data with significant variables as categorical data:
      ID  Months     DEATH       AGE SEX  CompositeStage LNInvolment  \
0      1      70 -1.026593 -0.588591   1        0.032170    1.604031
1      2      68 -1.026593 -0.588591   2       -2.174702   -0.623429
2      3      69 -1.026593 -0.422086   1       -1.071266   -0.623429
3      4      43  0.974096 -0.172330   2       -1.071266   -0.623429
4      5      71 -1.026593  0.993201   2        0.032170    1.604031
..   ...     ...       ...       ...  ..             ...         ...
338  339      65 -1.026593 -1.337860   1        0.032170    1.604031
339  340      61 -1.026593 -0.422086   1       -1.071266   -0.623429
340  341      65 -1.026593  0.327184   2       -1.071266   -0.623429
341  342      16  0.974096  1.159706   2        1.135606   -0.623429
342  343      31  0.974096  0.243931   2        1.135606    1.604031

     Comorbidity FamiliyHistoryOfCancer
0       0.913359                      0
1       0.913359                      0
2       0.913359                      0
3      -1.094860                      0
4       0.913359                      0
..           ...                    ...
338     0.913359                      0
339     0.913359                      0
340     0.913359                      0
341    -1.094860                      0
342    -1.094860                      0

[343 rows x 9 columns]
```

In [121…

```
cph_multivariate = CoxPHFitter(penalizer=0.1)
variables = ['Months', 'DEATH', 'AGE'] + [var for var, _ in significant_variables]
cph_multivariate.fit(buckley_james_data[variables], 'Months', 'DEATH', show_progress=True)
print(cph_multivariate.summary)
```

```
Iteration 1: norm_delta = 0.01879, step_size = 0.9500, log_lik = -1663.17959, newton_decrement = 0.06380, seconds_since_start =
0.0
Iteration 2: norm_delta = 0.00085, step_size = 0.9500, log_lik = -1663.11614, newton_decrement = 0.00013, seconds_since_start =
0.0
Iteration 3: norm_delta = 0.00004, step_size = 0.9500, log_lik = -1663.11600, newton_decrement = 0.00000, seconds_since_start =
0.1
Iteration 4: norm_delta = 0.00000, step_size = 1.0000, log_lik = -1663.11600, newton_decrement = 0.00000, seconds_since_start =
0.1
Convergence success after 4 iterations.
               coef  exp(coef)  se(coef)  coef lower 95%  coef upper 95%  \
covariate
AGE       -0.018672   0.981501  0.052274       -0.121127        0.083782

           exp(coef) lower 95%  exp(coef) upper 95%  cmp to         z  \
covariate
AGE                   0.885921             1.087392     0.0 -0.357205

                  p  -log2(p)
covariate
AGE        0.720938  0.472052
```

In [122…

```
n = len(buckley_james_data)
llf = cph_multivariate.log_likelihood_
k = cph_multivariate.params_.shape[0]
multivariate_aic = -2 * llf + 2 * k
multivariate_bic = -2 * llf + k * np.log(n)
print(cph_multivariate.summary)
```

```
                coef  exp(coef)  se(coef)  coef lower 95%  coef upper 95%  \
covariate
AGE        -0.018672   0.981501  0.052274       -0.121127        0.083782

           exp(coef) lower 95%  exp(coef) upper 95%  cmp to         z  \
covariate
AGE                   0.885921             1.087392     0.0 -0.357205

                  p  -log2(p)
covariate
AGE        0.720938  0.472052
```

In [123…

```python
# Print AIC and BIC for multivariate model
print("\nAIC value of the multivariate model:", multivariate_aic)
print("BIC value of the multivariate model:", multivariate_bic)

# Print AIC and BIC for univariate models
print("\nAIC and BIC for univariate models:")
for col, aic, bic in univariate_aic_bic:
    print(f"{col}: AIC={aic}, BIC={bic}")
```

```
AIC value of the multivariate model: 3328.2320093107332
BIC value of the multivariate model: 3332.0697397578992

AIC and BIC for univariate models:
DEATH: AIC=3328.0632186349508, BIC=3331.9009490821168
AGE: AIC=3328.0632186349508, BIC=3331.9009490821168
SEX: AIC=3328.0632186349508, BIC=3331.9009490821168
CompositeStage: AIC=3328.0632186349508, BIC=3331.9009490821168
LNInvolment: AIC=3328.0632186349508, BIC=3331.9009490821168
Comorbidity: AIC=3328.0632186349508, BIC=3331.9009490821168
FamiliyHistoryOfCancer: AIC=3328.0632186349508, BIC=3331.9009490821168
```

In [ ]: