Analysis on Retinopathy Treatment among Diabetic Patients

1 Introduction

Diabetic retinopathy is a diabetes complication that can cause vision loss and blindness with no symptoms or only mild vision problems at the early stages by affecting the blood vessels in the retina. The risk to develop eye complications increases the longer a patient has diabetes with less controlling of blood sugar, blood pressure and cholesterol. So a common treatment for diabetic retinopathy called the laser coagulation has been applied in order to delay diabetic retinopathy among patients with high risks of lost of visual acuity.

In this analysis, we first investigated the efficacy of two types of laser treatment, xenon and argon, on visual acuity and compared their efficacy to the control group. Then we studied the impact of the age at diagnosis and the clinic risk of diabetic retinopathy on visual acuity.

This analysis is intended to help the ophthalmologist at Michigan Medicine to determine and compare the potency of those two types of treatment. We found the xenon treatment is more effective than the argon treatment to prolong the time of vision loss of diabetic patients. Moreover, the clinic risk of diabetic retinopathy is a significant factor affecting visual acuity.

2 Methods

There are two main goals of our analysis. One is to quantify and compare the improvement for vision acuity by two different treatment types that were measured by the length of duration when participants lost their vision or were lost to follow-up (the lag-corrected time). Another is to measure how the patient's age of diagnosis and risk of diabetic retinopathy affect the visual acuity. To address our goals, we conducted survival analysis using Kaplan-Meier (KM) estimator and Cox Proportional Hazard (Cox PH) model. This aligned with the data and goals posed.

The experimental data were collected from patients with a high risk of acuity loss where one eye for each participant was assigned randomly by one randomized treatment type, either xenon or argon, and the eye without treatment was the control group. Both eyes of each participant were followed every three months and we had the record of the duration of vision loss or of the last follow-up. Hence, the data might be right censored due to no follow-up. This time-to-event analysis, or so-called survival analysis, can be used for analyzing the probability of participants surviving up (or losing vision acuity) to a given time.

The first step for survival analysis is to plot the curve of survival function by generating the KM curve. The KM estimator is a non-parametric method, meaning there is no assumption of the distribution of the outcome variable, that is the lag-corrected time in our analysis. This curve illustrates how the survival probability changes over the time.

In general, the survival probability of vision acuity reduces as time passes. That is the probability of loss vision increases over time. In our analysis, we focused on estimating the probability of loss vision other than the survival probability. To compare the efficacy of two treatment types, we drew the KM curve for each treatment group and the control group, and compared which curve of treatment type was significantly lower than the control group. In addition, we plotted the KM curve for different types of participants (adult/juvenile) and for participants with different risk of loss visual acuity.

KM estimator is a straightforward way to look at the survival/failure probability intuitively. However, only one variable is included and cannot solve the problem of incorporating covariates. Therefore, we introduce the Cox PH model to address our second question. This model assumes the independence of survival time between distinct individuals and a constant hazard ratio over time.

Note that in our data, each participant corresponds to two rows, one eye per row. Hence, we need to cluster our data in the Cox PH model. Moreover, there is a strong positive relationship between the participant's age at diagnosis and the type of participants (adult/juvenile). We dropped the 'type' variable when modelling to eliminate multicollinearity. After observing the data, we could find there is an interaction between the type of treatment and whether the eye is in treatment or in control. This should be included in our model. Moreover, there are several categorical variables and we used dummy coding to invert it to 0 or 1, where for example 1 means argon and 0 means xenon for the new dummy variable 'laser_argon'. Finally, the Cox PH model estimates the coefficient of covariates and we could see whether it was statistically significant or not at the 95% confidence level.

3 Results

3.1 Overview of the Data

The data set contains 394 rows collected from 197 clinic patients, where one patient corresponds to two rows, one in the treatment group and one in the control group. There are 9 variables we could use later. 'id' was used in the modeling part in order to deal with the clustered data. We dropped the type of participants (adult/juvenile) in the modelling part considering the multicollinearity, but we used this variable in the KM curve to observe the difference in survival probability between the two groups. The remaining variables: types of treatment (xenon/argon), which eye was recorded (left/right), the age at diagnosis, under treatment or control, lag-corrected time to loss of vision or to the last follow-up, status at the lag-corrected time (loss of vision/no follow-up), and the clinic risk to loss of acuity. In addition, there is no missing value in this data set. For the clinic risk, we assume that the larger the number is the higher the risk is.

Below is the baseline table that displays the median and inner quartile range (pr percentage of categorical variables) for the explanatory variables.

Variable	Median (IQR) or Percentage
Age at diagnosis	16(10, 30)
Time to vision loss or no follow-up (months)	38.8(13.98, 54.25)
Clinic risk to vision loss	10(9, 11)
Xenon treatment (%)	50.76
Right eye treated (%)	45.18
Adult (%)	42.13
Loss of vision (%)	39.34
Treatment group (%)	50

Table 1: Baseline Table

3.2 Explanatory Data Analysis

We first have a look at the distribution of lag-corrected time among different groups shown in Figure 1. We can find the density curve of the treatment group has shifted to the right in overall compared to the curve of the control group. This indicates the patients in the treatment group in general have a longer time until they lose their visions than those in the control group.

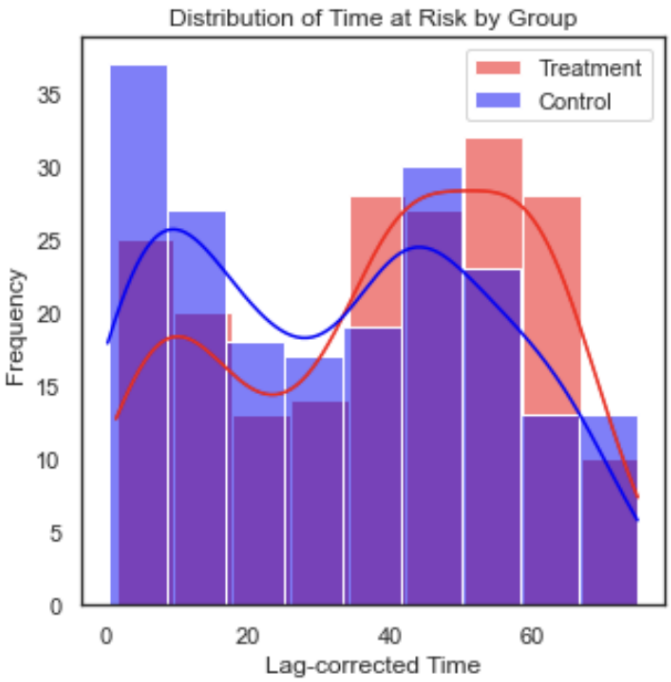


Figure 1: Comparing distributions of time to loss of vision or to no follow-up between treatment group and control group

Then we investigate the relationship between each variable and the lag-corrected time to loss of vision or to the last follow-up as shown in Figure 2. We can find the argon treatment seems to have more efficacy than the xenon treatment compared to the control group. Moreover, there is no difference between the right or left eye for the treatment in general. We also cannot tell the difference among people of different ages. In addition, laser treatment can

improve the condition of almost all patients with different risks of losing acuity, but with a risk of 8.

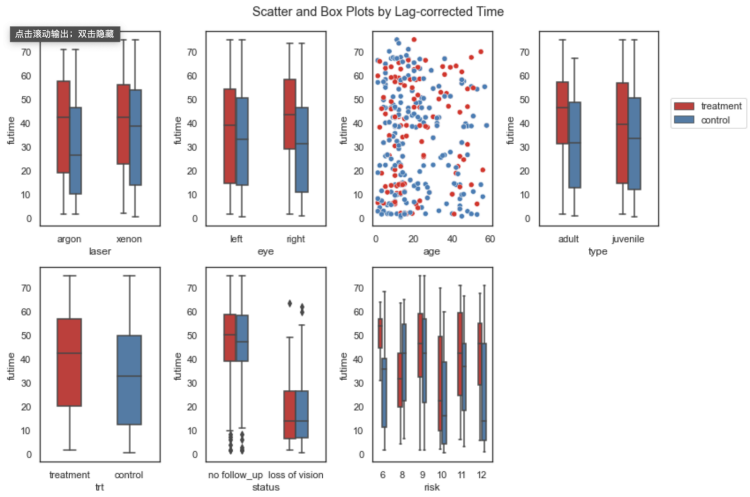


Figure 2: Scatter and box plots by the lag-corrected time between the treatment group and the control group

3.3 Results and Analysis

First, we drew the Kaplan-Meier curve to estimate the probability of loss of vision over time in Figure 3. The curve above other curves indicates this group has a higher probability of vision loss. Therefore, we can find that the xenon treatment has slightly higher efficacy than the argon treatment, and obviously both treatments delay the time until loss of vision. The second plot suggests that people with a higher risk of diabetic retinopathy are more likely to lose vision. In addition, there is no obvious difference in treatment between the adults and juveniles.

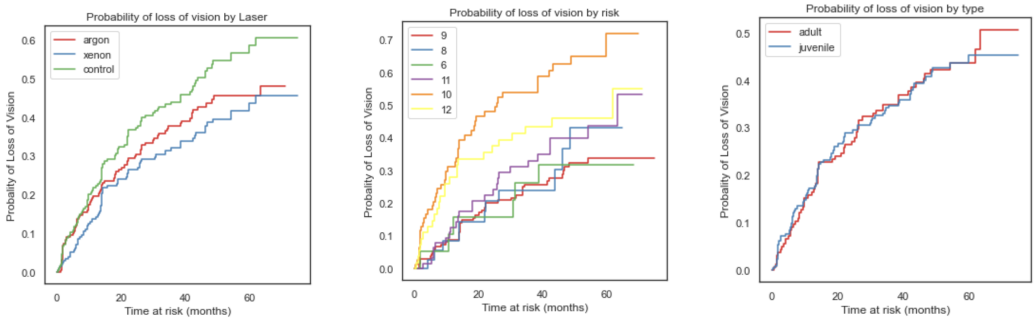


Figure 3: KM curve for different treatment types, risks, and age types

In order to corporate covariates, we used the Cox PH model to study how the age at diagnosis and the clinic risk of diabetic retinopathy affect visual acuity. The outcome variable is the time until vision loss or the last follow-up. Also, there is an interaction between the treatment type and whether the eye is in the treatment or control group. The age type variable has been dropped due to multicollinearity. Table 2 displays the estimated coefficients and the p-value correspondingly.

Variable	exp(Coefficient)	P-value	Confidence Interval
Age at diagnosis	1.01	0.32	(0.99, 1.02)
Right eye treated	0.81	0.23	(0.57, 1.15)
Xenon treatment	0.86	0.43	(0.58, 1.26)
Risk of diabetic retinopathy	1.15	0.02	(1.03, 1.29)
In the treatment group	0.46	<0.005	(0.31, 0.70)
Interaction	0.97	0.92	(0.54, 1.74)

Table 2: Coefficients with Interaction term

We can find that whether the eye is treated or not and the risk of diabetic retinopathy is statistically significant to influence the time when patients will lose their vision, while the age of patients at diagnosis and treatment type is not significant at the 95% confidence level. This contradicts to what we find in the KM curve that the xenon treatment has slightly more efficacy than the argon.

For one increase in the risk of diabetic retinopathy, the patient's hazard rate to get blindness will increase by 15%. Moreover, the probability of the eye not being treated to lose vision is 54 % higher than that of eyes treated.

4 Conclusion

In our analysis, we utilized the KM estimator to determine the efficacy of two different treatment types and applied the Cox PH model to study the potential impact of patient's age at diagnosis and the risk of loss of acuity on visual acuity. We find that patients who did not have treatment is more likely to lose their vision by 54%. Every increase in the clinic risk of diabetic retinopathy will cause an increase in the probability of vision loss by 15%. Moreover, the age at diagnosis is not significant to make a difference in the time patients got blindness.

We dropped the variable of age type to eliminate multicollinearity. There might be a better choice to deal with it. We could explore more about the data set and check the variance inflation factor to measure the amount of multicollinearity.

appendix

September 28, 2022

1 Load Data

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
import statsmodels.formula.api as smf
from lifelines import KaplanMeierFitter
from lifelines import CoxPHFitter
import seaborn as sns
```

[32]:
```
df = pd.read_csv('./diabeticVision.csv', index_col=0)
df.head(6)
```

[32]:

	id	laser	eye	age	type	trt	futime	status	risk
1	5	argon	left	28	adult	1	46.23	0	9
2	5	argon	left	28	adult	0	46.23	0	9
3	14	argon	right	12	juvenile	1	42.50	0	8
4	14	argon	right	12	juvenile	0	31.30	1	6
5	16	xenon	right	9	juvenile	1	42.27	0	11
6	16	xenon	right	9	juvenile	0	42.27	0	11

[33]:
```
df.shape
```

[33]:
```
(394, 9)
```

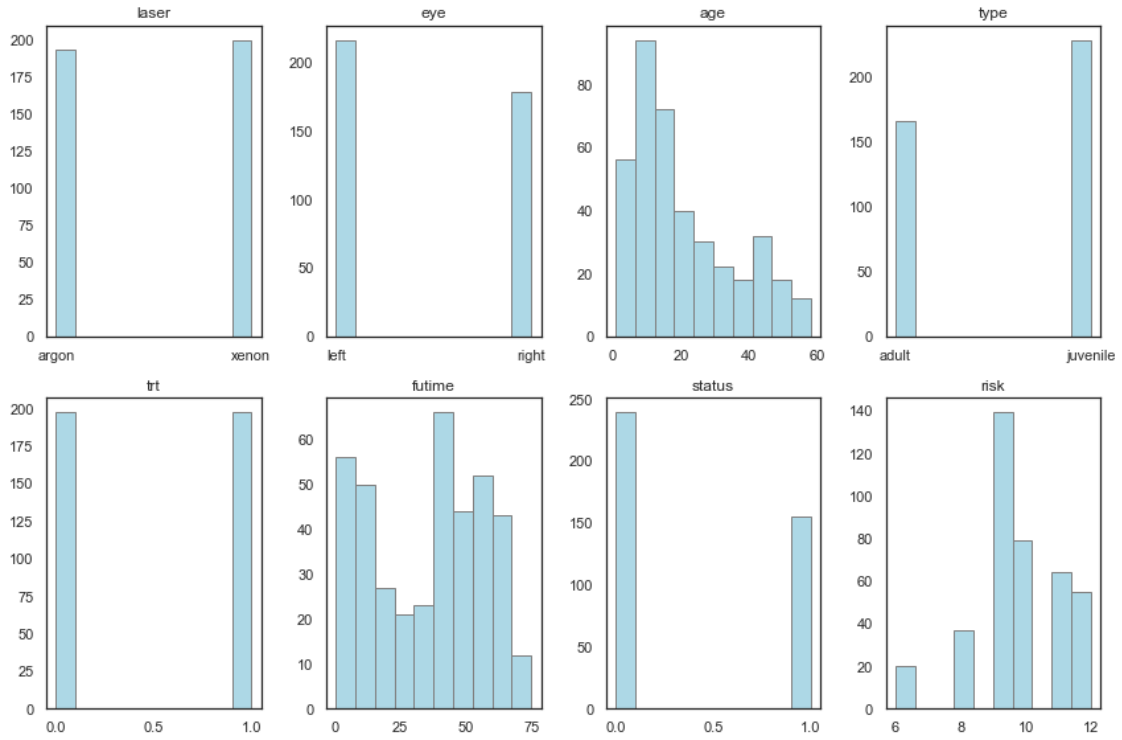
[88]:
```
df['risk'].unique()
```

[88]:
```
array([ 9,  8,  6, 11, 10, 12])
```

2 EDA

[34]:
```
# histogram
plt.rcParams['figure.figsize'] = [12, 8]
def draw_histogram(df, variables, n_rows, n_cols):
    fig = plt.figure()
    for i, var_name in enumerate(variables):
```

```
        ax = fig.add_subplot(n_rows,n_cols,i+1)
        df[var_name].hist(ax=ax, color='lightblue', ec='grey')
        ax.set_title(var_name)
        ax.grid(False)
    fig.tight_layout()
    plt.show()

draw_histogram(df.iloc[:,1:], df.iloc[:,1:].columns, 2, 4)
```



[35]:
```
# check null value
df.isna().sum()
```

[35]:
```
id         0
laser      0
eye        0
age        0
type       0
trt        0
futime     0
status     0
risk       0
dtype: int64
```

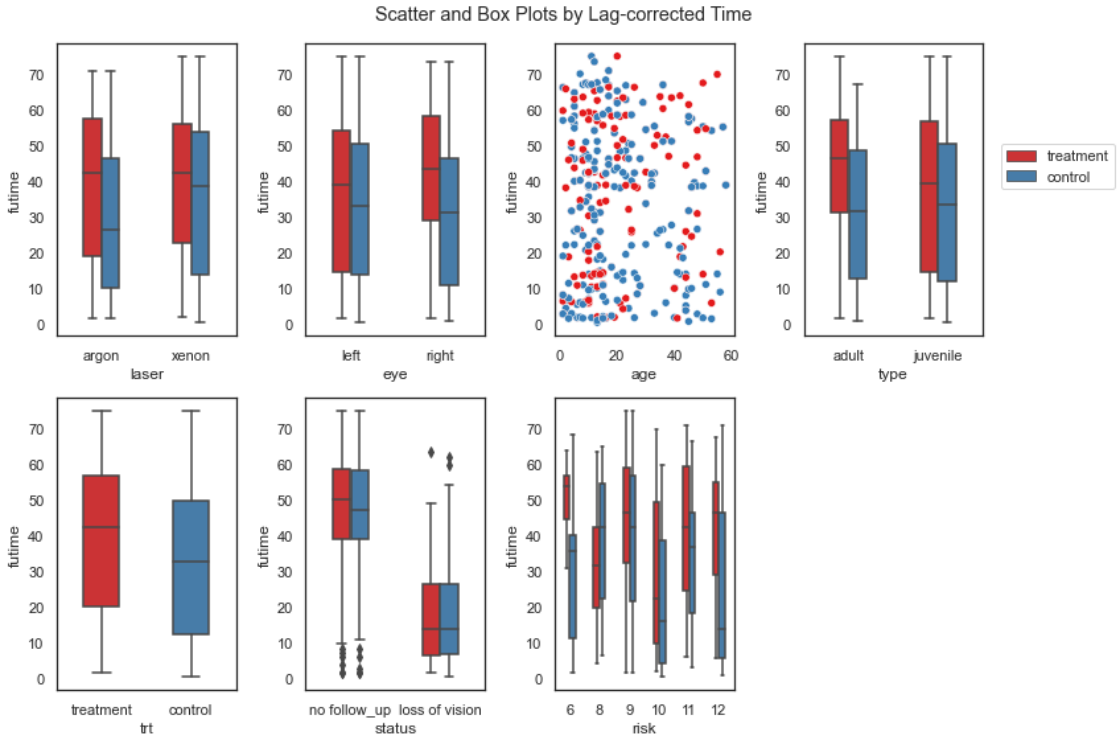
[36]:
```
df[df['type']=='adult'].sort_values(by=['age']).head(5)
# we can find if age <20, then type is juvenile
```

[36]:

	id	laser	eye	age	type	trt	futime	status	risk
352	1562	xenon	right	20	adult	0	6.57	1	9
331	1487	argon	right	20	adult	1	62.37	0	9
332	1487	argon	right	20	adult	0	43.70	1	8
187	800	xenon	right	20	adult	1	65.23	0	9
242	1029	xenon	right	20	adult	0	13.37	1	12

[66]:
```
df1 = df.copy()
df1['trt'] = df1['trt'].map({0: 'control', 1:'treatment'})
df1['status'] = df1['status'].map({0: 'no follow_up', 1:'loss of vision'})
```

[79]:
```
# boxplot to check outliers
def draw_boxplot_bygroup(df, outcome, n_rows, n_cols, group):
    fig = plt.figure()
    variables = df.columns.drop(outcome)
    sns.set(style='white', palette='Set1')
    for i, var_name in enumerate(variables):
        ax = fig.add_subplot(n_rows, n_cols, i+1)
        if var_name == 'age':
            sns.scatterplot(x=var_name, y=outcome, hue=group, data=df,␣
 ↪legend=False, ax=ax)
        elif var_name == 'trt':
            sns.boxplot(x=var_name, y=outcome, data=df, width=0.4, ax=ax)
        elif var_name == 'type':
            sns.boxplot(x=var_name, y=outcome, hue=group, data=df, width=0.4,␣
 ↪ax=ax)
            ax.legend(loc=(1.1, 0.5))
        else:
            sns.boxplot(x=var_name, y=outcome, hue=group, data=df, width=0.4,␣
 ↪ax=ax)
            ax.legend_.remove()
    fig.suptitle('Scatter and Box Plots by Lag-corrected Time')
    fig.tight_layout()
    plt.show()

draw_boxplot_bygroup(df1.iloc[:,1:], 'futime', 2, 4, 'trt')
```



2.1 Pairwise Correlation

[82]:
```
df1.iloc[:,1:].corr()
```

```
/var/folders/qx/wlpqnbz178962z833f_yq5gh0000gn/T/ipykernel_53541/2031927840.py:1
: FutureWarning: The default value of numeric_only in DataFrame.corr is
deprecated. In a future version, it will default to False. Select only valid
columns or specify the value of numeric_only to silence this warning.
  df1.iloc[:,1:].corr()
```

[82]:

	age	futime	risk
age	1.000000	-0.002297	-0.044486
futime	-0.002297	1.000000	-0.109343
risk	-0.044486	-0.109343	1.000000

[85]:
```
# get categorical variable
cat_vars = df.select_dtypes(exclude=np.number)
df_num = df.copy()
for var in cat_vars:
    if var == 'laser':
        df_num[var] = df[var].map({'xenon':0, 'argon':1})
    elif var == 'eye':
        df_num[var] = df[var].map({'left':0, 'right':1})
    else:
```

```
        df_num[var] = df[var].map({'juvenile':0, 'adult':1})
df_num.iloc[:,1:].corr().round(4)
```

[85]:

	laser	eye	age	type	trt	futime	status	risk
laser	1.0000	-0.0576	-0.0877	-0.0795	-0.0000	-0.0858	0.0279	0.0262
eye	-0.0576	1.0000	0.0982	0.0930	-0.0000	0.0183	-0.1047	-0.1220
age	-0.0877	0.0982	1.0000	0.8375	-0.0000	-0.0023	0.0361	-0.0445
type	-0.0795	0.0930	0.8375	1.0000	0.0000	0.0369	0.0284	-0.0379
trt	-0.0000	-0.0000	-0.0000	0.0000	1.0000	0.1543	-0.2442	-0.0362
futime	-0.0858	0.0183	-0.0023	0.0369	0.1543	1.0000	-0.6375	-0.1093
status	0.0279	-0.1047	0.0361	0.0284	-0.2442	-0.6375	1.0000	0.1228
risk	0.0262	-0.1220	-0.0445	-0.0379	-0.0362	-0.1093	0.1228	1.0000

We can see there is a strong positive correlation between `age` and `type`.
And a strong negative correlation between `status` and `futime`.
So we remove `type` for our modelling part.

2.2 Baseline Table

[13]:
```
df.iloc[:,1:].describe().round(2).transpose()
```

[13]:

	count	mean	std	min	25%	50%	75%	max
age	394.0	20.78	14.81	1.0	10.00	16.0	30.00	58.00
trt	394.0	0.50	0.50	0.0	0.00	0.5	1.00	1.00
futime	394.0	35.58	21.36	0.3	13.98	38.8	54.25	74.97
status	394.0	0.39	0.49	0.0	0.00	0.0	1.00	1.00
risk	394.0	9.70	1.48	6.0	9.00	10.0	11.00	12.00

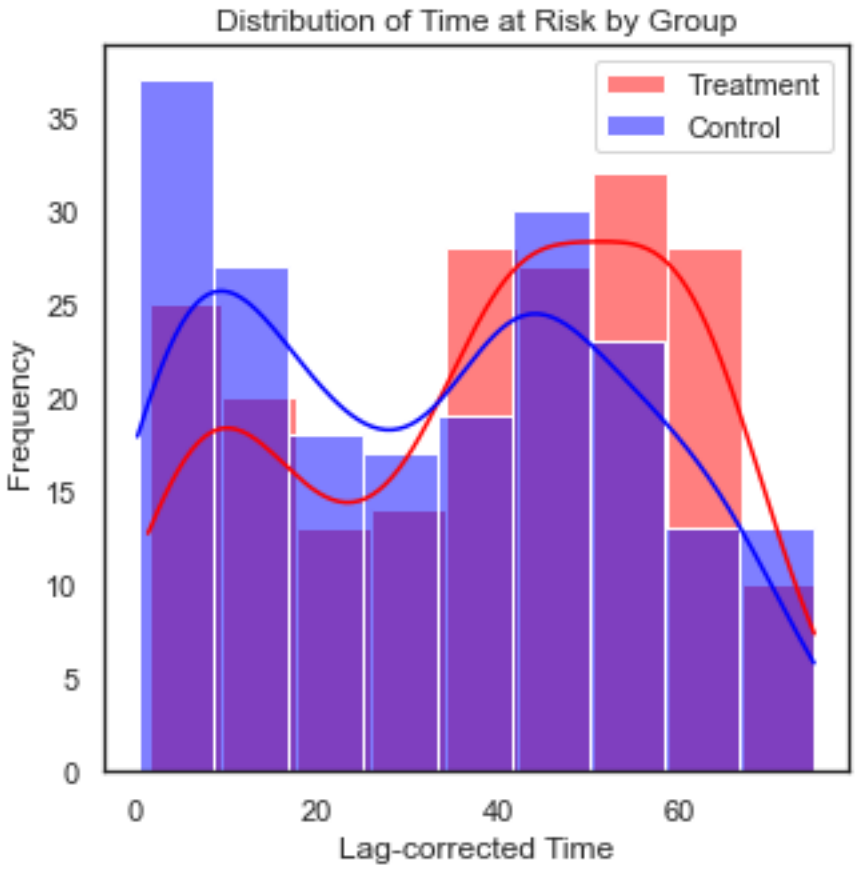
[90]:
```
def get_categorical_percentages(df):
    df_cat = df.select_dtypes(exclude=np.number)
    for var in df_cat.columns:
        perc = df[var].value_counts() / df[var].count()
        print(var)
        print(perc)

get_categorical_percentages(df1)
```

```
laser
xenon    0.507614
argon    0.492386
Name: laser, dtype: float64
eye
left     0.548223
right    0.451777
Name: eye, dtype: float64
type
juvenile    0.57868
adult       0.42132
Name: type, dtype: float64
```

```
trt
treatment    0.5
control      0.5
Name: trt, dtype: float64
status
no follow_up      0.606599
loss of vision    0.393401
Name: status, dtype: float64
```

2.3 Variable Distribution

[98]:
```
plt.rcParams['figure.figsize'] = [5, 5]
ax = sns.histplot(df_num[df_num.trt==1]['futime'], label='Treatment', kde=True,␣
 ↪color='red')
sns.histplot(df_num[df_num.trt==0]['futime'], label='Control', kde=True,␣
 ↪color='blue')
ax.set(xlabel='Lag-corrected Time', ylabel='Frequency')
plt.title('Distribution of Time at Risk by Group')
plt.legend()
plt.show()
```



[92]:
```
df_num.head(10)
```

[92]:

	id	laser	eye	age	type	trt	futime	status	risk
1	5	1	0	28	1	1	46.23	0	9
2	5	1	0	28	1	0	46.23	0	9
3	14	1	1	12	0	1	42.50	0	8
4	14	1	1	12	0	0	31.30	1	6
5	16	0	1	9	0	1	42.27	0	11
6	16	0	1	9	0	0	42.27	0	11
7	25	1	0	9	0	1	20.60	0	11
8	25	1	0	9	0	0	20.60	0	11
9	29	0	0	13	0	1	38.77	0	9
10	29	0	0	13	0	0	0.30	1	10

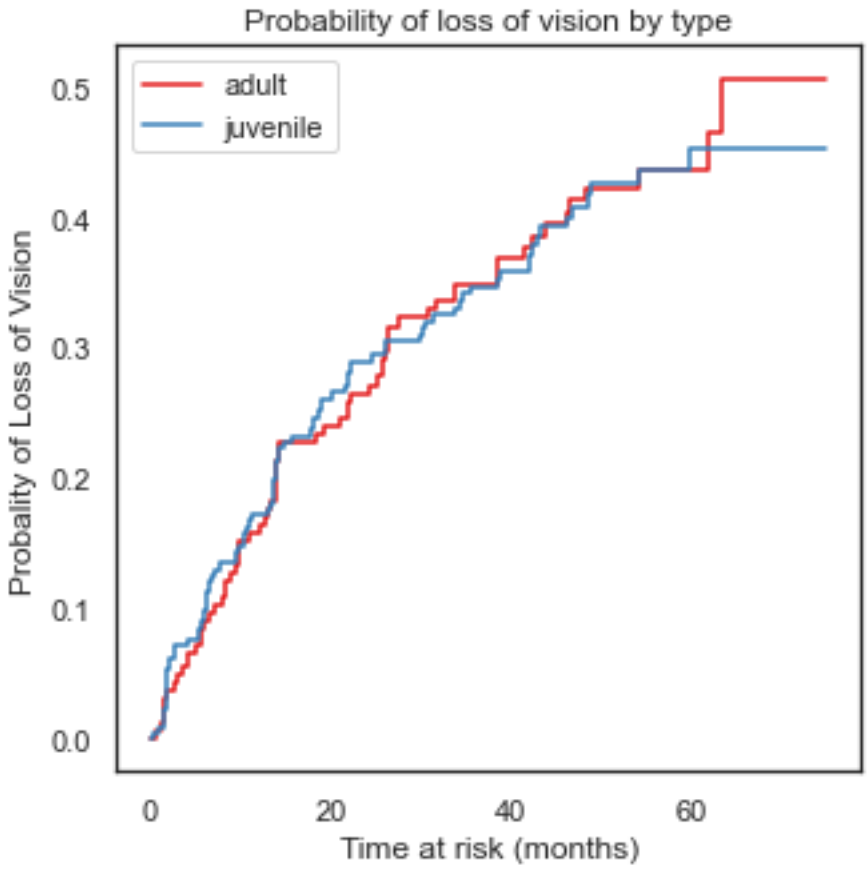
3 Survival Analysis

3.1 Kaplan-Maier Curve Estimation

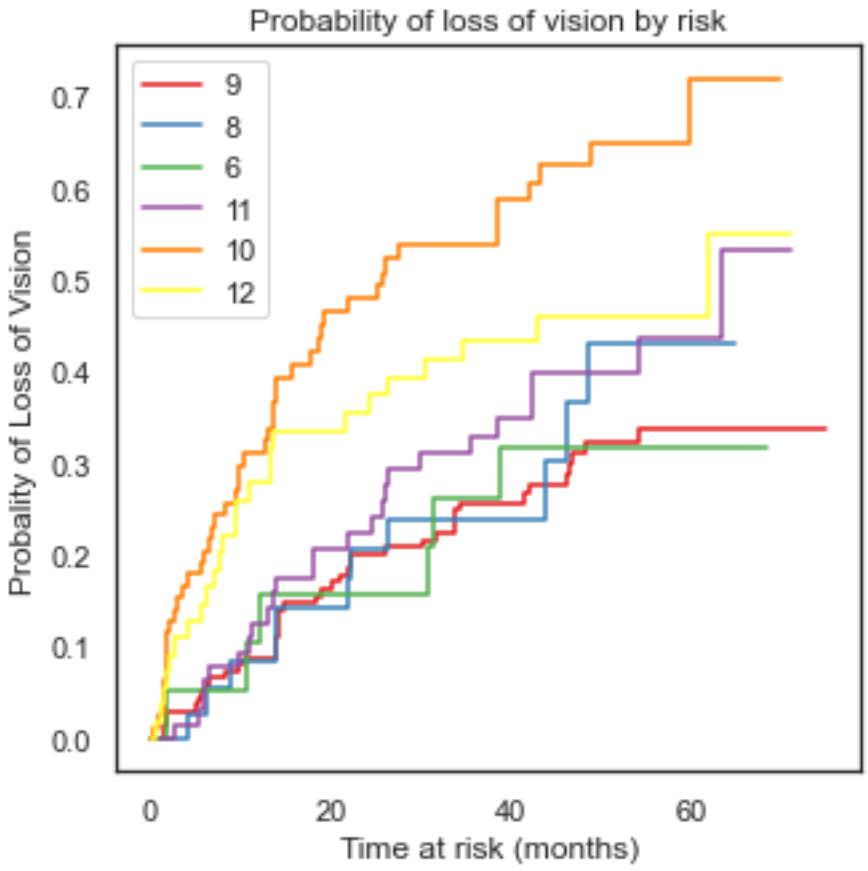
reference
More examples - The curve illustrates how the survival probabilities changes over the time horizon.
- As time passes, it is more likely to have a loss of vision.

[124]:
```
def plot_km_curve_bygroup(df,group):
    ax = plt.subplot(111)
    kmf = KaplanMeierFitter()
    for v in df[group].unique():
        kmf.fit(durations=df[df[group]==v].futime,␣
↪event_observed=df[df[group]==v].status, label=str(v))
        kmf.plot_cumulative_density(ci_show=False)

    plt.title('Probability of loss of vision by ' + group)
    plt.xlabel('Time at risk (months)')
    plt.ylabel('Probality of Loss of Vision')
    ax.legend(loc='upper left')
```

[129]:
```
plot_km_curve_bygroup(df, 'type')
```

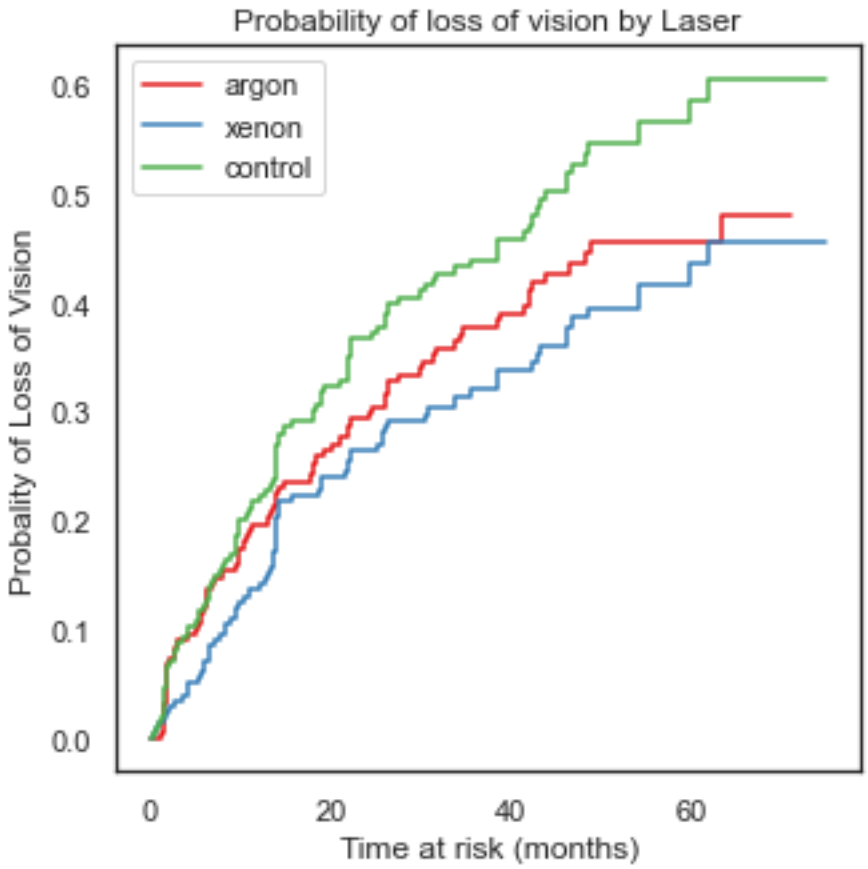


[128]:
```
plot_km_curve_bygroup(df, 'risk')
```



[131]:
```
# for laser type
ax = plt.subplot(111)
kmf = KaplanMeierFitter()
group = 'laser'
for v in df[group].unique():
    kmf.fit(durations=df[df[group]==v].futime, event_observed=df[df[group]==v].
 ↪status, label=str(v))
    kmf.plot_cumulative_density(ci_show=False)
kmf = KaplanMeierFitter()
kmf.fit(durations=df[df['trt']==0].futime, event_observed=df[df['trt']==0].
 ↪status, label='control')
kmf.plot_cumulative_density(ci_show=False)
plt.title('Probability of loss of vision by Laser')
plt.xlabel('Time at risk (months)')
plt.ylabel('Probality of Loss of Vision')
ax.legend(loc='upper left')
```

[131]: <matplotlib.legend.Legend at 0x7f9f70fc2560>



3.2 Cox Proportional Hazard Model

documentation

[138]:
```
# one-hot encoding for categorical variable
df2 = df.drop('type', axis=1)
df_dummies = pd.get_dummies(df2, drop_first=True)
df_dummies
```

[138]:

	id	age	trt	futime	status	risk	laser_xenon	eye_right
1	5	28	1	46.23	0	9	0	0
2	5	28	0	46.23	0	9	0	0
3	14	12	1	42.50	0	8	0	1
4	14	12	0	31.30	1	6	0	1
5	16	9	1	42.27	0	11	1	1
..	…	…	…	…	…	…	…	…
390	1727	33	0	2.90	1	10	0	1
391	1746	3	1	45.90	0	10	0	1
392	1746	3	0	1.43	1	10	0	1
393	1749	32	1	41.93	0	9	0	1

```
394  1749   32    0   41.93       0     9             0          1

[394 rows x 8 columns]
```

[139]:
```
# Model the Hazard
cph = CoxPHFitter(alpha=0.05) # 95% ci
cph.fit(df=df_dummies, duration_col='futime', event_col='status',␣
 ↪cluster_col='id',
       formula='age + trt + risk + laser_xenon + eye_right + trt*laser_xenon')
cph.print_summary()
```

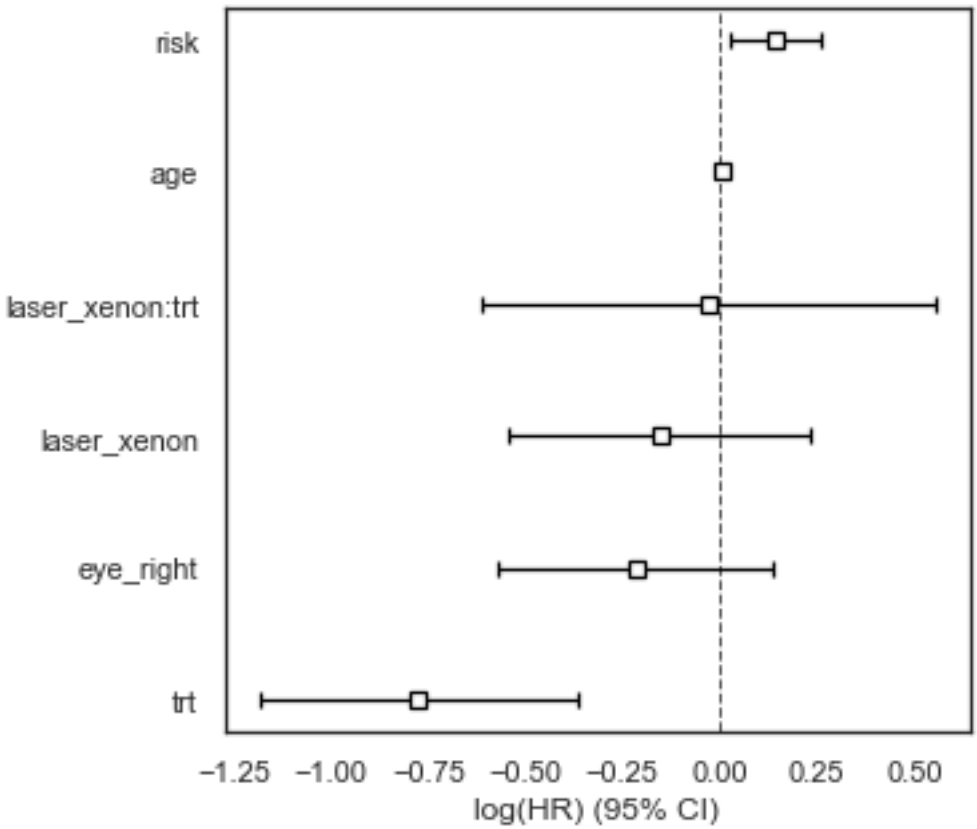
```
/opt/anaconda3/envs/504/lib/python3.10/site-
packages/lifelines/utils/__init__.py:997: FutureWarning: iteritems is deprecated
and will be removed in a future version. Use .items instead.
  nonnumeric_cols = [col for (col, dtype) in df.dtypes.iteritems() if dtype.name
== "category" or dtype.kind not in "biuf"]
/opt/anaconda3/envs/504/lib/python3.10/site-
packages/lifelines/utils/printer.py:62: FutureWarning: In future versions
`DataFrame.to_latex` is expected to utilise the base implementation of
`Styler.to_latex` for formatting and rendering. The arguments signature may
therefore change. It is recommended instead to use `DataFrame.style.to_latex`
which also contains additional functionality.
  return summary_df[columns].to_latex(float_format="%." + str(self.decimals) +
"f")
```

covariate	coef	exp(coef)	se(coef)	coef lower 95%	coef upper 95%	exp(coef) lower 95%	exp(co
age	0.01	1.01	0.01	-0.01	0.02	0.99	
eye_right	-0.22	0.81	0.18	-0.57	0.14	0.57	
laser_xenon	-0.16	0.86	0.20	-0.54	0.23	0.58	
risk	0.14	1.15	0.06	0.03	0.26	1.03	
trt	-0.77	0.46	0.21	-1.18	-0.36	0.31	
laser_xenon:trt	-0.03	0.97	0.30	-0.61	0.55	0.54	

3.2.1 Plot Coefficients

[141]:
```
cph.plot()
```

[141]:
```
<AxesSubplot:xlabel='log(HR) (95% CI)'>
```

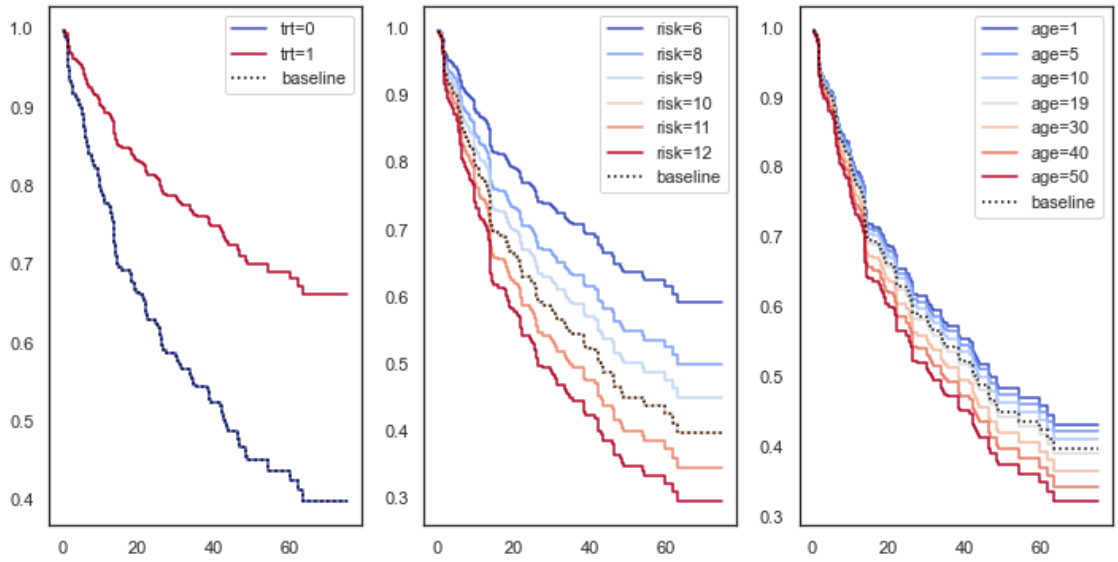


3.2.2 Plot Partial Effects

We can see how the survival changes as we change the covariate values.

[152]:
```
fig, axes = plt.subplots(1,3,figsize=(12,6))
cph.plot_partial_effects_on_outcome(covariates = 'trt', values = [0,1], cmap =␣
↪'coolwarm', ax=axes[0])
cph.plot_partial_effects_on_outcome(covariates = 'risk', values =␣
↪[6,8,9,10,11,12], cmap = 'coolwarm'
                                    , ax=axes[1])
cph.plot_partial_effects_on_outcome(covariates = 'age', values =␣
↪[1,5,10,19,30,40,50], cmap = 'coolwarm'
                                    , ax=axes[2])
```

[152]: <AxesSubplot:>



3.2.3 Cheak Proportional Hazard Assumption

[154]:
```
from lifelines.statistics import proportional_hazard_test
results = proportional_hazard_test(cph, df_dummies, time_transform='rank')
results.print_summary(decimals=3, model="untransformed variables")
```

```
/opt/anaconda3/envs/504/lib/python3.10/site-
packages/lifelines/utils/__init__.py:997: FutureWarning: iteritems is deprecated
and will be removed in a future version. Use .items instead.
  nonnumeric_cols = [col for (col, dtype) in df.dtypes.iteritems() if dtype.name
== "category" or dtype.kind not in "biuf"]
/opt/anaconda3/envs/504/lib/python3.10/site-
packages/lifelines/statistics.py:143: FutureWarning: In future versions
`DataFrame.to_latex` is expected to utilise the base implementation of
`Styler.to_latex` for formatting and rendering. The arguments signature may
therefore change. It is recommended instead to use `DataFrame.style.to_latex`
which also contains additional functionality.
  return self.summary.to_latex()
```

	test_statistic	p	-log2(p)
age	0.355188	0.551191	0.859376
eye_right	1.121886	0.289513	1.788301
laser_xenon	0.990842	0.319537	1.645946
laser_xenon:trt	0.359913	0.548554	0.866293
risk	1.729660	0.188454	2.407714
trt	0.028136	0.866790	0.206245

None violates the assumption.