Brian Tenneson

ADS 534

08/23/2024

Statement regarding academic integrity: Large Language Models (LLMs) were utilized alongside human oversight to perform data analysis and generate insights across various contexts and datasets.

1

1(a)

Placebo Group Times: [0. 1. 2. 3. 4. 5. 6. 7. 9. 10. (it continues to) 59.]

Placebo Group Survival Probabilities: [1. 0.9787234 0.89172577 0.73947991 0.73947991 0.69466294 0.64984598 0.6274375 0.58096064 0.55772222 (it continues to) 0.28486113]

Treatment Group Times: [1. 2. 3. 4. 5. 6. 9. 10. 13. 17. (it continues to) 59.]

Treatment Group Survival Probabilities: [0.94736842 0.83591331 0.80804954 0.75232198 0.7244582 0.66873065 0.66873065 0.66873065 0.66873065 0.66873065 0.60504202 (it continues to) 0.44642857].

Placebo Group:

S(5): 0.695 (69.5% probability of survival beyond 5 months)

S(10): 0.558 (55.8% probability of survival beyond 10 months)

S(25): 0.404 (40.4% probability of survival beyond 25 months)

Treatment Group:

S(5): 0.724 (72.4% probability of survival beyond 5 months)

S(10): 0.669 (66.9% probability of survival beyond 10 months)

S(25): 0.536 (53.6% probability of survival beyond 25 months)

We used the following python code to arrive at those results:

```
# Function to find Kaplan-Meier estimates at specific time points
def km_estimate_at_times(times_array, survival_prob_array, specific_times):
    # Dictionary to store survival probabilities at specified times
    estimates = {}
    for t in specific_times:
        # Find the last time that is less than or equal to 't' and use its survival probab
        idx = np.where(times_array <= t)[0][-1]
        estimates[t] = survival_prob_array[idx]
    return estimates

# Specific times to evaluate the survival probabilities
specific_times = [5, 10, 25]

# Getting estimates for placebo and treatment groups at specific times
km_estimates_placebo = km_estimate_at_times(times_placebo, survival_prob_placebo, specific_
km_estimates_treatment = km_estimate_at_times(times_treatment, survival_prob_treatment, sp

km_estimates_placebo, km_estimates_treatment
```

The Kaplan-Meier estimates of $S(10)$ provide a statistical measure of the probability that patients will survive without a recurrence of their condition—tumor recurrence in this case—for at least 10 months after treatment. Let's delve into the results for both the placebo group and the treatment group at $S(10)$:

Placebo Group:

Kaplan-Meier Estimate at

S(10): 0.558

Interpretation: This estimate suggests that there is a 55.8% chance that a patient in the placebo group will not experience tumor recurrence within the first 10 months following the start of the observation period. The lower survival probability reflects the natural course of the illness without the intervention of the active drug, indicating a significant risk of recurrence during this time.

Treatment Group:

Kaplan-Meier Estimate at

S(10): 0.669

Interpretation: For the treatment group, the estimate shows a 66.9% probability that a patient will remain tumor-free for at least 10 months. This higher survival probability compared to the placebo group indicates the potential effectiveness of the treatment in delaying or preventing tumor recurrence. The treatment seems to provide a protective effect, enhancing the patients' chances of a longer recurrence-free period.

1(b)

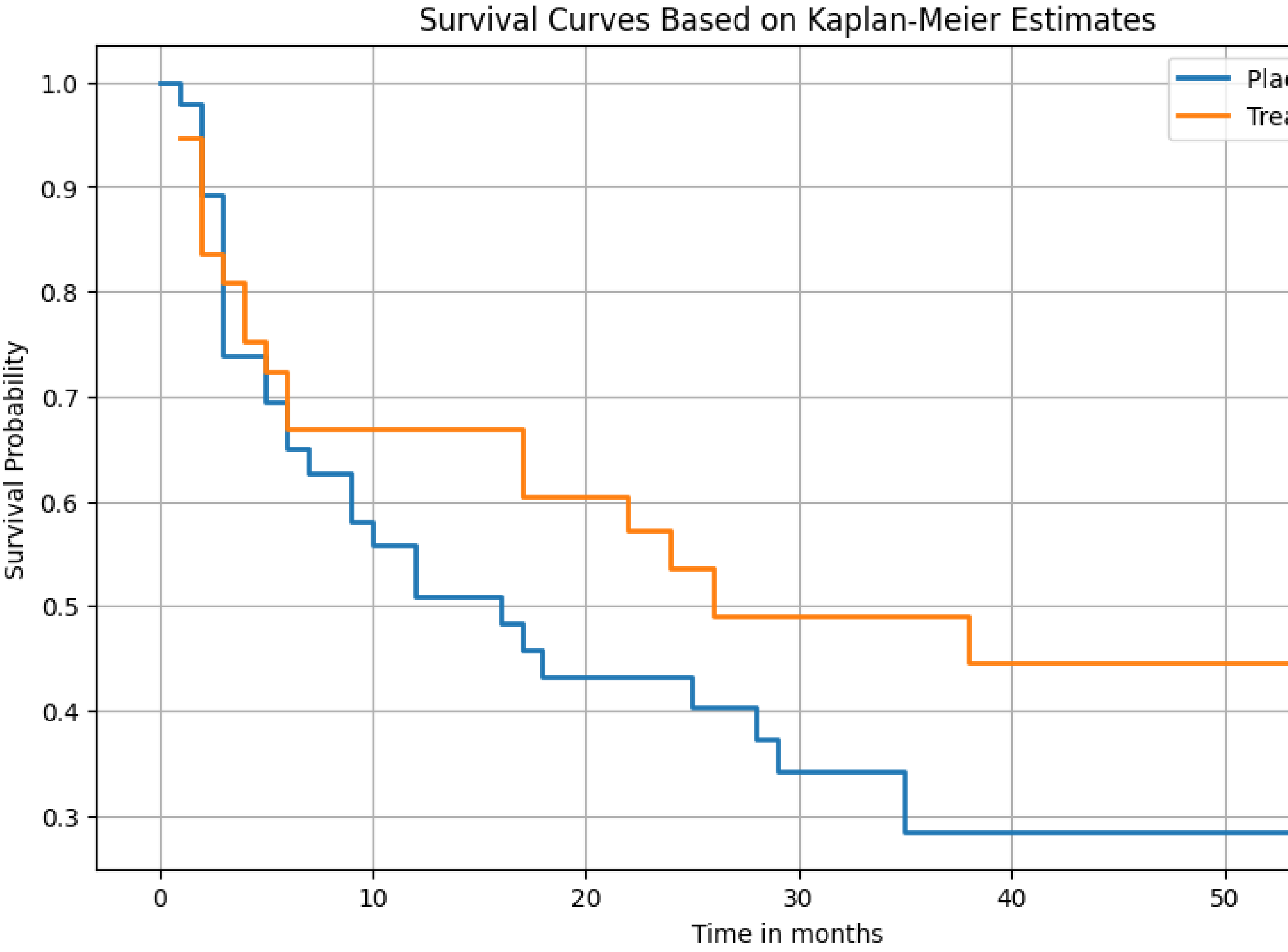
To construct the plot, I used the following python code:

```
import matplotlib.pyplot as plt

# Plotting the survival curves based on Kaplan-Meier estimates
plt.figure(figsize=(10, 6))
plt.step(times_placebo, survival_prob_placebo, where="post", label="Placebo Group", linewid
plt.step(times_treatment, survival_prob_treatment, where="post", label="Treatment Group", 

# Adding details to the plot
plt.title("Survival Curves Based on Kaplan-Meier Estimates")
plt.xlabel("Time in months")
plt.ylabel("Survival Probability")
plt.legend()
plt.grid(True)

# Show plot
plt.show()
```



It looks like the curve corresponding to the treatment group dominates the group getting the placebo, meaning that the treatment is effective at raising the probability of survival compared to the placebo and this difference becomes more pronounced as time progresses.

1(c)

To test the null hypothesis that the distributions of recurrence times are identical in the two treatment groups, the Log-Rank Test is the most suitable method. This test is widely used in survival analysis to compare the survival distributions of two or more groups.

Log-Rank Test Overview:

Purpose: It is used to test whether there are statistically significant differences between the survival curves of two groups.

Null Hypothesis (H_0): The survival functions of the two groups are the same across all time points.

Alternative Hypothesis (H_1): The survival functions of the two groups differ at least at one time point.

I used the following python code to obtain the result that follows:

```
import pandas as pd
import numpy as np
import scipy.stats

# Reload the dataset
new_data_path = "G:\\My Drive\\Summer - 2 -2024\\ADS 534\\homework 8\\bladder.csv"
new_bladder_data = pd.read_csv(new_data_path)

# Segmenting the data
new_placebo_group = new_bladder_data[new_bladder_data['group'] == 0]
new_treatment_group = new_bladder_data[new_bladder_data['group'] != 0]

# Define the log-rank test function
def log_rank_test(data1, data2):
    # Combine the data and sort by time
    combined_data = pd.concat([data1, data2])
    combined_data.sort_values('time', inplace=True)
```

```
# Calculate observed and expected deaths
observed1 = data1[data1['censor'] == 1].shape[0]
observed2 = data2[data2['censor'] == 1].shape[0]
expected1, expected2 = 0, 0
total_observed = 0

for time in combined_data['time'].unique():
    n1 = len(data1[(data1['time'] >= time)])
    n2 = len(data2[(data2['time'] >= time)])
    n = n1 + n2

    d = len(combined_data[(combined_data['time'] == time) & (combined_data['censor'] ==

    e1 = d * (n1 / n)
    e2 = d * (n2 / n)

    expected1 += e1
    expected2 += e2

    total_observed += d

# Calculate the test statistic
test_stat = (observed1 - expected1)**2 / expected1 + (observed2 - expected2)**2 / expe
p_value = 1 - scipy.stats.chi2.cdf(test_stat, 1)  # 1 degree of freedom
```

```
    return test_stat, p_value


# Perform the Log–Rank Test on the new data
log_rank_stat, log_rank_p_value = log_rank_test(new_placebo_group, new_treatment_group)


log_rank_stat, log_rank_p_value
```

Result:

(1.4272903673467985, 0.23220716145916098)

Given that the p-value exceeds 0.05, this is greater than the typical significance level of 0.05. Thus, we fail to reject the null hypothesis, indicating that there is not enough statistical evidence to conclude that there are significant differences in the survival distributions of recurrence times between the two treatment groups. The results suggest that the treatment does not significantly alter the time to tumor recurrence compared to the placebo, based on the data provided. (I find this to be odd given prior conclusions such as what the graph tells us.)

1(d)

The hazard function $h(t)$ for a subject at time t can be expressed as:

$$h(t \mid X) = h_0(t)\exp(\beta_1 X_1 + \beta_2 X_2)$$

where:

$h(t \mid X)$ is the hazard at time t given covariates X

$h_0(t)$ is the baseline hazard, representing the hazard for a person with baseline values of zero for all covariates.

X_1 and X_2represent the explanatory variables, with X_1 being the group and X_2 being the number.

β_1 and β_2 are the coefficients for the covariates, which measure the effect size of each covariate on the hazard. These coefficients are estimated from the data.

Group (X_1): Indicates the treatment group, which could be a binary indicator where 1 represents treatment and 0 represents placebo.

Number (X_2): Represents the number of tumors initially removed, where we might code this as a continuous variable or categorize it (e.g., 1 for a single tumor, 2 for two or more tumors).

The exponential term $\exp(\beta_1 X_1 + \beta_2 X_2)$ is the hazard ratio, describing how the hazard changes with a one-unit change in the covariate while holding other covariates constant. A β value greater than 0 indicates an increase in hazard (and thus a decrease in survival time) for each unit increase in the predictor. Conversely, a β value less than 0 suggests a decrease in hazard (and thus an increase in survival time).

1(e)

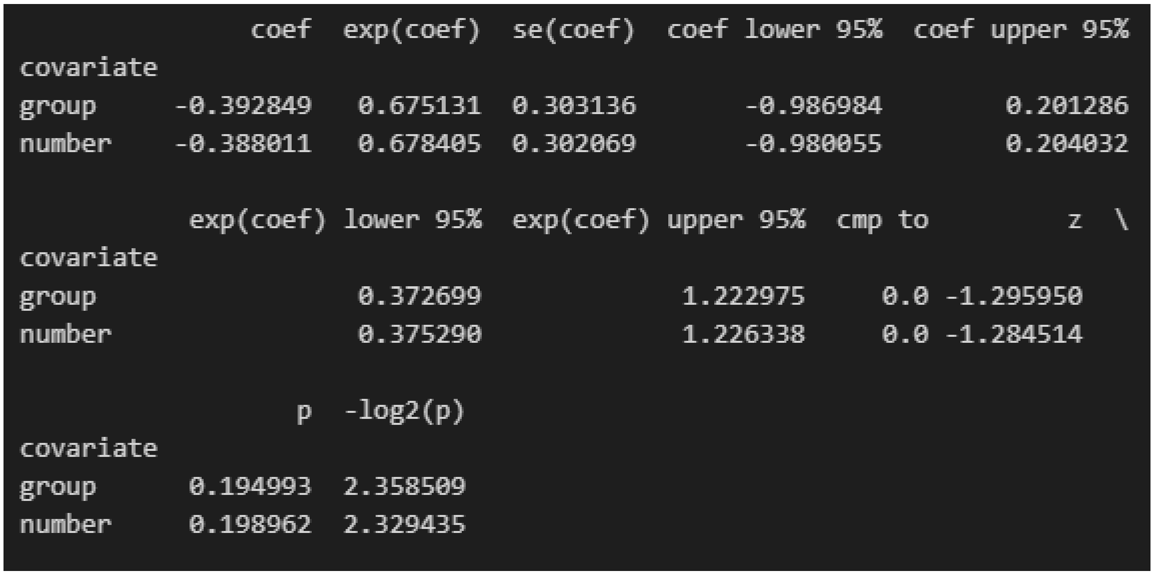
Using the following python code, we fit a hazard model whose results follow:

```
import pandas as pd
from lifelines import CoxPHFitter

# Load your data
data_path = "G:\\My Drive\\Summer - 2 -2024\\ADS 534\\homework 8\\bladder.csv"
data = pd.read_csv(data_path)

# Instantiate the Cox Proportional Hazards model
cph = CoxPHFitter()

# Fit the model
cph.fit(data[['time', 'censor', 'group', 'number']], duration_col='time', event_col='censor
# Print the summary of the model to see the coefficients and statistics
print(cph.summary)
```



```
                coef  exp(coef)  se(coef)  coef lower 95%  coef upper 95%
covariate
group      -0.392849   0.675131  0.303136       -0.986984        0.201286
number     -0.388011   0.678405  0.302069       -0.980055        0.204032

           exp(coef) lower 95%  exp(coef) upper 95%  cmp to         z  \
covariate
group                 0.372699             1.222975     0.0 -1.295950
number                0.375290             1.226338     0.0 -1.284514

                  p  -log2(p)
covariate
group      0.194993  2.358509
number     0.198962  2.329435
```

Confidence Interval of Hazard Ratio=[$e^{Lower\ Bound\ of\ \beta}$,$e^{Upper\ Bound\ of\ \beta}$]

Coefficient for group:

$\beta = -0.392849$

Standard Error:

$SE = 0.303136$

Calculating the Confidence Interval of β:

Lower Bound of $\beta = -0.392849 - 1.96 \times 0.303136 \approx -0.987$

Upper Bound of $\beta = -0.392849 + 1.96 \times 0.303136 \approx 0.201$

Converting to the Hazard Ratio Confidence Interval:

Lower Bound of Hazard Ratio

$e^{-0.987} \approx 0.373$

Upper Bound of Hazard Ratio

$e^{0.201} \approx 1.223$

These calculations yield a 95% confidence interval for the hazard ratio of approximately [0.373, 1.223], which I identified in the output.

Interpretation:

This method shows how uncertainty in the coefficient estimate translates into uncertainty in the hazard ratio.

The inclusion of 1 within this interval suggests that the effect of the treatment may not be statistically significant, as previously discussed.