ENRE 447 - 0101
Homework 04:

Due on March 10th, 2025 at 03:30 PM

Dr. Groth, 03:30 PM

Vai Srivastava

March 10, 2025

Problem 1:

Nine light bulbs are observed, and the exact failure time of each is recorded as 70, 150, 250, 360, 485, 650, 855, 1130, and 1540. Estimate and plot the following:

1. Cdf of failure times
2. Pdf of failure times
3. Reliability function
4. Hazard-rate function

Solution

$$\hat{F}(t) = \frac{1}{9}\sum_{i=1}^{9} \mathbf{1}\{t_i \le t\},$$

$$\hat{f}(t) \approx \text{KDE estimate of } f(t),$$

$$\hat{R}(t) = 1 - \hat{F}(t),$$

$$\hat{h}(t) = \frac{\hat{f}(t)}{\hat{R}(t)}. \qquad \square$$

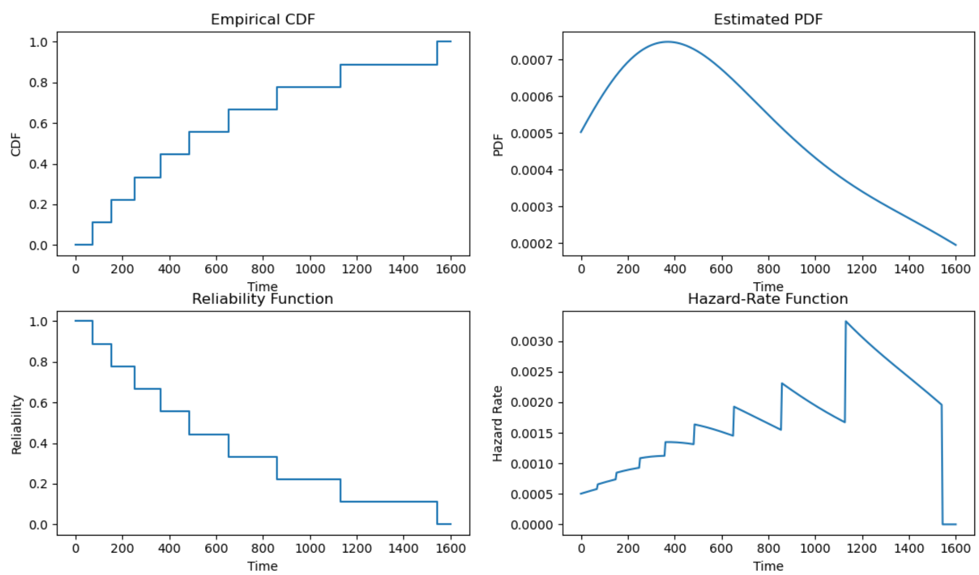

Figure 1: Plot for HW04 P01

Code

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import gaussian_kde

```

Problem 1 continued on next page. . .

```python
data = np.array([70, 150, 250, 360, 485, 650, 855, 1130, 1540])

t_grid = np.linspace(0, 1600, 400)

cdf_emp = np.array([np.sum(data <= t) for t in t_grid]) / len(data)

kde = gaussian_kde(data)
pdf_est = kde(t_grid)

R_est = 1 - cdf_emp

h_est = np.divide(pdf_est, R_est, out=np.zeros_like(pdf_est), where=R_est>0)

plt.figure(figsize=(10, 8))

plt.subplot(2,2,1)
plt.step(t_grid, cdf_emp, where='post')
plt.xlabel('Time')
plt.ylabel('CDF')
plt.title('Empirical CDF')

plt.subplot(2,2,2)
plt.plot(t_grid, pdf_est)
plt.xlabel('Time')
plt.ylabel('PDF')
plt.title('Estimated PDF')

plt.subplot(2,2,3)
plt.step(t_grid, R_est, where='post')
plt.xlabel('Time')
plt.ylabel('Reliability')
plt.title('Reliability Function')

plt.subplot(2,2,4)
plt.plot(t_grid, h_est)
plt.xlabel('Time')
plt.ylabel('Hazard Rate')
plt.title('Hazard-Rate Function')

plt.tight_layout()
plt.show()
```

Listing 1: Python code for HW04 P01

Problem 2:

The following time to failure data are found when 158 transformer units are put under test. No failures are observed prior to 1750 hours.

Time Interval:	$1750 \to 2250$	$2250 \to 2750$	$2750 \to 3250$	$3250 \to 3750$	$3750 \to 4250$	$4250 \to 4750$
# Failures:	17	54	27	17	19	24

Use a nonparametric method to estimate $\mathrm{f}(t), \mathrm{h}(t)$, and $\mathrm{R}(t)$ of the transformers.

Solution

Define:

$$\begin{aligned} &n_1 = 158, \quad n_2 = 158 - 17 = 141, \quad n_3 = 141 - 54 = 87, \\ &n_4 = 87 - 27 = 60, \quad n_5 = 60 - 17 = 43, \quad n_6 = 43 - 19 = 24, \\ &\Delta t = 500 \text{ (hours)}. \end{aligned}$$

The piecewise constant density is estimated by

$$\hat{f}(t) = \begin{cases} \frac{17}{158 \cdot 500}, & 1750 \le t < 2250, \\ \frac{54}{141 \cdot 500}, & 2250 \le t < 2750, \\ \frac{27}{87 \cdot 500}, & 2750 \le t < 3250, \\ \frac{17}{60 \cdot 500}, & 3250 \le t < 3750, \\ \frac{19}{43 \cdot 500}, & 3750 \le t < 4250, \\ \frac{24}{24 \cdot 500}, & 4250 \le t < 4750, \end{cases}$$

$$\hat{R}(t) = \begin{cases} 1, & 1750 \le t < 2250, \\ \frac{141}{158}, & 2250 \le t < 2750, \\ \frac{87}{158}, & 2750 \le t < 3250, \\ \frac{60}{158}, & 3250 \le t < 3750, \\ \frac{43}{158}, & 3750 \le t < 4250, \\ \frac{24}{158}, & 4250 \le t < 4750, \\ 0, & t \ge 4750, \end{cases}$$

$$\hat{h}(t) = \frac{\hat{f}(t)}{\hat{R}(t)}. \qquad \square$$

Problem 3:

Time to failure data from eight devices placed on an accelerated test is shown below.

i:	1	2	3	4	5	6	7	8
TTF:	65	85	90	95	340	405	555	575

1. Use probability plotting to estimate the parameter of the exponential distribution.
2. Discuss the suitability of the explonential distribution for this data.

Solution

$$\bar{t} = \frac{65 + 85 + 90 + 95 + 340 + 405 + 555 + 575}{8} = 276.25,$$

$$\hat{\lambda} = \frac{1}{\bar{t}} \approx \frac{1}{276.25} \approx 0.00362,$$

$$\text{Exponential CDF:} \quad F(t) = 1 - e^{-\lambda t},$$

$$\ln(-\ln(1 - F(t))) = \ln(\lambda) + \ln t.$$

A probability plot of $\ln(-\ln(1 - \hat{F}(t)))$ versus $\ln t$ should be linear with slope 1.

Thus, $\hat{\lambda} \approx 0.00362$.

(Discussion: Deviations from linearity indicate that the exponential model may not be fully suitable.) □

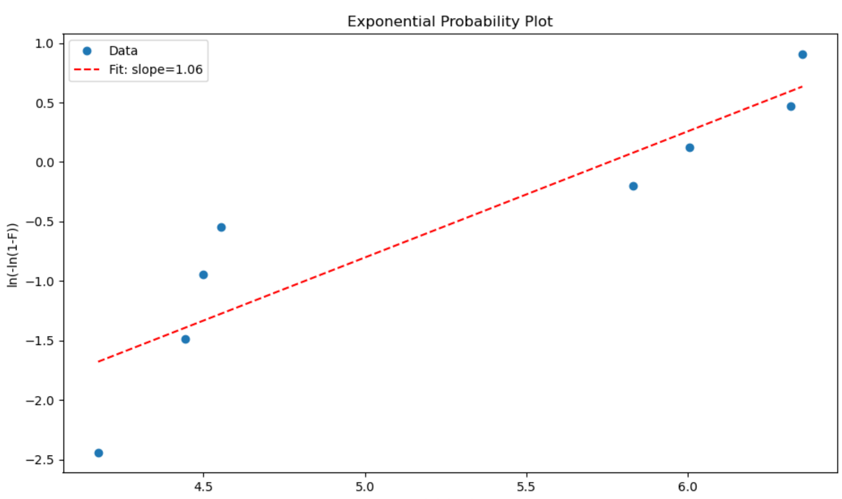

Figure 2: Plot for HW04 P03

Code

```python
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as st

# Failure times data
data = np.array([65, 85, 90, 95, 340, 405, 555, 575])
n = len(data)

# Estimate lambda by sample mean
lambda_hat = 1.0 / np.mean(data)

# Use median ranks for empirical CDF
sorted_data = np.sort(data)
# Using (i-0.3)/(n+0.4) as median rank estimates
F_emp = (np.arange(1, n+1) - 0.3) / (n + 0.4)

# Transform for exponential probability plot: Y = ln(-ln(1-F))
Y = np.log(-np.log(1 - F_emp))
X = np.log(sorted_data)

# Linear regression to check linearity
slope, intercept, r_value, p_value, std_err = st.linregress(X, Y)

plt.figure(figsize=(8,6))
plt.plot(X, Y, 'o', label='Data')
plt.plot(X, intercept + slope*X, 'r--', label=f'Fit: slope={slope:.2f}')
plt.xlabel('ln(Time)')
plt.ylabel('ln(-ln(1-F))')
plt.title('Exponential Probability Plot')
plt.legend()
plt.show()

print("Estimated lambda =", lambda_hat)
```

Listing 2: Python code for HW04 P03

Problem 4:

In an accelerated test, failures of 8 units are recorded after 8, 17, 21, 21, 22, 39, 42, and 47 days after starting in operation. The other two units are operating after 50 days.

1. Perform a Weibull probability plot of these data.
2. Find the parameters of the Weibull distribution from the plot.

Solution

Using only the 8 failures, assign median rank estimates:

$$\hat{F}(t_{(i)}) \approx \frac{i - 0.3}{8 + 0.4}, \quad i = 1, \ldots, 8,$$

$$\ln\big[-\ln(1 - \hat{F}(t_{(i)}))\big] = \beta \, \ln(t_{(i)}) - \beta \, \ln(\eta).$$

A linear regression of

$$Y_i = \ln\big[-\ln(1 - \hat{F}(t_{(i)}))\big] \quad \text{versus} \quad X_i = \ln(t_{(i)})$$

yields slope $\hat{\beta}$ and intercept $-\hat{\beta}\ln(\eta)$. From the plot one obtains, approximately,

$$\hat{\beta} \approx 1.82 \qquad \square$$

$$\hat{\eta} \approx 31.5 \qquad \square$$

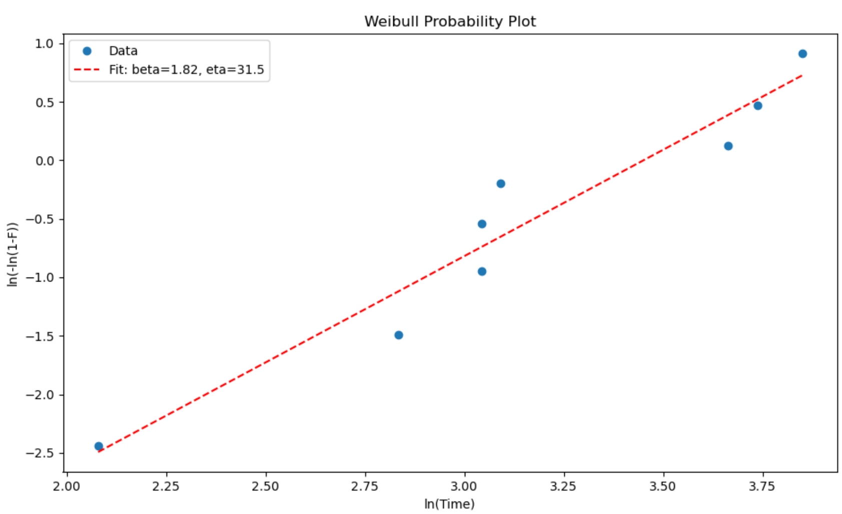

Figure 3: Plot for HW04 P04

Code

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as st

# Failure times (uncensored)
```

Problem 4 continued on next page. . .

```python
failures = np.array([8, 17, 21, 21, 22, 39, 42, 47])
n_fail = len(failures)

# Compute median ranks using (i - 0.3)/(n + 0.4)
i = np.arange(1, n_fail+1)
F_est = (i - 0.3) / (n_fail + 0.4)

# Transform data for Weibull probability plot
X = np.log(failures)
Y = np.log(-np.log(1 - F_est))

# Linear regression to estimate Weibull parameters
slope, intercept, r_value, p_value, std_err = st.linregress(X, Y)
beta_hat = slope
eta_hat = np.exp(-intercept / beta_hat)

plt.figure(figsize=(8,6))
plt.plot(X, Y, 'o', label='Data')
plt.plot(X, intercept + slope*X, 'r--', label=f'Fit: beta={beta_hat:.2f}, eta={eta_hat:.1f}')
plt.xlabel('ln(Time)')
plt.ylabel('ln(-ln(1-F))')
plt.title('Weibull Probability Plot')
plt.legend()
plt.show()

print("Estimated beta =", beta_hat)
print("Estimated eta =", eta_hat)
```

Listing 3: Python code for HW04 P04