PART B: Take-Home Coding Final Exam

Survival Analysis in Actuarial Science

Dr. Adhitya Ronnie Effendie, M.Sc.

General Information

- **Course**: Data Analytic For Actuarial Science (DAFAS) 2025
- **Level**: Graduate Students
- **Programming Language**: Python
- **Submission**: Jupyter Notebook (`.ipynb`) or Python script (`.py`) + short written interpretation (PDF)
- **Deadline**: Friday, 19 December 2025 from 07.00 pm to 09.00 pm via GitHub Classroom

Context

You are provided with a synthetic insurance portfolio containing life insurance policies observed over time. Some policies experience a claim event (death/TPD/CI), while others are **right-censored** due to administrative censoring or lapse. In addition, policies may enter the observation window late, implying **left truncation (delayed entry)**.

Your task is to analyze this portfolio using standard and advanced survival analysis techniques and interpret the results from an actuarial perspective.

Datasets

You are given two CSV files:

1. `survival_policies.csv`

Main survival dataset containing both censored and uncensored observations.

- `policy_id`: unique policy identifier
- `entry_month`: delayed entry time (months since study start)
- `duration_months`: observed survival time since entry
- `event`: 1 if claim occurred, 0 if right-censored

- `age_at_entry`: age at policy entry
- `gender`: `M` or `F`
- `smoker`: 1 if smoker, 0 otherwise
- `product_type`: TERM10, TERM20, WHOLE
- `region_risk`: Low, Medium, High
- `sum_assured_k`: sum assured (in thousands)
- `annual_premium_k`: annual premium (in thousands)
- `split`: train/test indicator

2. `survival_claims.csv`

Claim-level dataset (only for policies with event = 1).

- `policy_id`: policy identifier
- `duration_months`: time to claim since entry
- `cause`: Death, TPD, or CI
- `claim_amount_k`: claim amount paid (in thousands)
- `notification_lag_days`: reporting delay
- `claim_month`: calendar month of claim occurrence

Allowed Python Packages

You may use the following libraries:

- `numpy`
- `pandas`
- `matplotlib`
- `scipy`
- `lifelines`

No other survival-analysis-specific libraries are allowed without prior approval.

Task A: Data Audit and Feature Preparation (Medium)

A1. Perform basic data validation:

- Check for missing or invalid values
- Verify that survival times are positive
- Compute overall and segmented censoring rates

A2. Prepare covariates:

- Center age at 40
- Encode categorical variables using dummy variables
- Create at least one interaction term (e.g. smoker × high-risk region)

Task B: Nonparametric Survival Analysis (Medium)

B1. Estimate Kaplan–Meier survival curves accounting for left truncation.

B2. Plot survival curves:

- Overall portfolio
- Smoker vs non-smoker

B3. Conduct a log-rank test comparing smokers and non-smokers.

B4. Interpret the results in an actuarial context (risk selection, underwriting).

Task C: Cox Proportional Hazards Model (Difficult)

C1. Fit a Cox proportional hazards model on the training dataset:

- Include delayed entry (left truncation)
- Use relevant demographic and policy covariates

C2. Report hazard ratios with 95% confidence intervals.

C3. Evaluate predictive performance using concordance index on train and test sets.

C4. Check proportional hazards assumptions and propose a remedy if violated.

Task D: Parametric Survival Model (Difficult)

D1. Fit a Weibull parametric survival model (AFT or PH).

D2. Compare the parametric model with the Cox model using likelihood-based criteria.

D3. Plot predicted survival curves for at least two contrasting risk profiles.

Task E: Actuarial Application – Net Single Premium (Difficult)

Assume a term insurance benefit equal to the sum assured, payable at the moment of event. Let the annual effective interest rate be $i = 4\%$.

E1. Approximate the Net Single Premium (NSP) using monthly discretization:

$$\text{NSP} \approx \sum_{m=1}^{120} v^{m/12} \Pr(T \in (m-1, m]) \times B, \quad v = (1+i)^{-1}.$$

E2. Compute NSP for policies in the test set using your fitted parametric model.

E3. Summarize NSP by risk group (e.g. smoker vs non-smoker).

E4. Provide a high-level comparison between NSP and observed annual premium.

Submission Requirements

- Well-documented Python code
- Clear plots and tables
- A concise actuarial interpretation (1–2 pages)
- Explicit discussion of assumptions and limitations

Assessment Criteria

Component	Weight
Data preparation and audit	15%
Nonparametric analysis	20%
Cox model and diagnostics	30%
Parametric modeling	20%
Actuarial interpretation (NSP)	15%

Academic Integrity

This is an individual take-home assignment. Discussion of high-level concepts is allowed, but all code and interpretations must be your own.