

INSTITUTE OF ACTUARIES OF INDIA

CS2B - Risk Modelling and Survival Analysis

May 2026 Examination

INDICATIVE SOLUTION

Introduction:

The indicative solution has been written by the Examiners with the aim of helping candidates. The solutions given are only indicative. It is realized that there could be other points as valid answers and examiners have given credit for any alternative approach or interpretation which they consider to be reasonable.

Solution.1**(i) Data Import and recoding**

```
library(dplyr)

data <- read.csv("C:\\Users\\Desktop\\CS2B\\employee_attrition.csv")

str(data)

# Convert to factors

dataAttrition <- factor(dataAttrition)

dataDepartment <- factor(dataDepartment)

dataOverTime <- factor(dataOverTime)

dataEducationField <- factor(dataEducationField)
```

[2]

(ii)**Checking for Missing Values and Handling them**

```
# Check missing values
```

```
colSums(is.na(data))
```

Output:

Age	0
Department	0
MonthlyIncome	5
YearsAtCompany	0
JobSatisfaction	2
OverTime	0
NumCompaniesWorked	0
EducationField	0
Attrition	0

There are 5 missing values in Monthly Income and 2 missing values in job satisfaction.

Handling Missing Values

```
dataMonthlyIncome[is.na(dataMonthlyIncome)] <- median(dataMonthlyIncome,
na.rm=TRUE)
```

```
dataJobSatisfaction[is.na](http://is.na)(dataJobSatisfaction)] <- median(data$JobSatisfaction, na.rm=TRUE)
```

After running this code:

Missing values in Monthly Income will be replaced by (highlighted below):

	Attrition	Age	Department	MonthlyIncome	YearsAtCompany
1	0	44	Tech	25713	12
2	1	31	HR	76178	8
3	1	28	Sales	69407	14
4	0	56	HR	90340	14
5	0	58	Sales	70990	10
6	0	40	Sales	95672	15
7	0	55	Sales	77108	16
8	1	45	Sales	85545	3
9	0	55	Sales	104791	0
10	1	45	Tech	78141	7
11	0	49	HR	43793	16

Missing values in Job satisfaction will be replaced by (highlighted below):

	Attrition	Age	Department	MonthlyIncome	YearsAtCompany	JobSatisfaction	OverTime
1	0	44	Tech	25713	12	3	No
2	1	31	HR	76178	8	3	No
3	1	28	Sales	69407	14	1	Yes
4	0	56	HR	90340	14	2	No
5	0	58	Sales	70990	10	4	No
6	0	40	Sales	95672	15	2	No
7	0	55	Sales	77108	16	2	Yes
8	1	45	Sales	85545	3	2	Yes
9	0	55	Sales	104791	0	1	No
10	1	45	Tech	78141	7	3	Yes
11	0	49	HR	43793	16	2	No

[2]

(iii) Relationship between Attrition and Overtime /Department/Education Field (Strongly associated variables)

Attrition vs OverTime

```
table(dataOverTime, dataAttrition)
```

Like this:

OverTime	0	1
No	74	28
Yes	66	32

Calculating row wise proportion

```
prop.table(table(dataOverTime, dataAttrition), 1)
```

OverTime	0	1
No	0.7254902	0.2745098
Yes	0.6734694	0.3265306

Attrition vs **Department**

```
table(dataDepartment, dataAttrition)
```

Output:

Department	0	1
HR	62	23
Sales	40	17
Tech	38	20

Calculating row wise proportion

```
prop.table(table(dataDepartment, dataAttrition), 1)
```

Department	0	1
HR	0.7294118	0.2705882
Sales	0.7017544	0.2982456
Tech	0.6551724	0.3448276

#Attrition vs **EducationField**

```
table(dataEducationField, dataAttrition)
```

Education Field	0	1
Engineering	36	15
Finance	40	12
HR	32	18
Marketing	32	15

Row-wise proportion

```
prop.table(table(dataEducationField, dataAttrition), 1)
```

Education Field	0	1
Engineering	0.7058824	0.2941176
Finance	0.7692308	0.2307692
HR	0.6400000	0.3600000
Marketing	0.6808511	0.3191489

[3]

(iv)

Interpretation**Overtime****Interpretation:**

- Employees working **OverTime = Yes** have a **higher proportion of attrition**
- Employees with **no overtime** show relatively lower attrition

Conclusion:

OverTime is strongly associated with Attrition and is an important predictor.

Department**Interpretation:**

- **Tech Department** show highest attrition rates followed by Sales
- Others (e.g., HR) may have comparatively lowest attrition

Conclusion:

Department is another key variable influencing Attrition

Interpretation:

Attrition rates vary across different education fields

HR shows relatively higher attrition (18 out of 50 $\approx$ 36%)

Engineering and Marketing also show moderate attrition levels

Finance has the lowest attrition rate (12 out of 52 $\approx$ 23%)

Conclusion:

Education Field is an important factor influencing Attrition, with employees from HR and technical/marketing backgrounds showing higher likelihood of leaving compared to Finance.

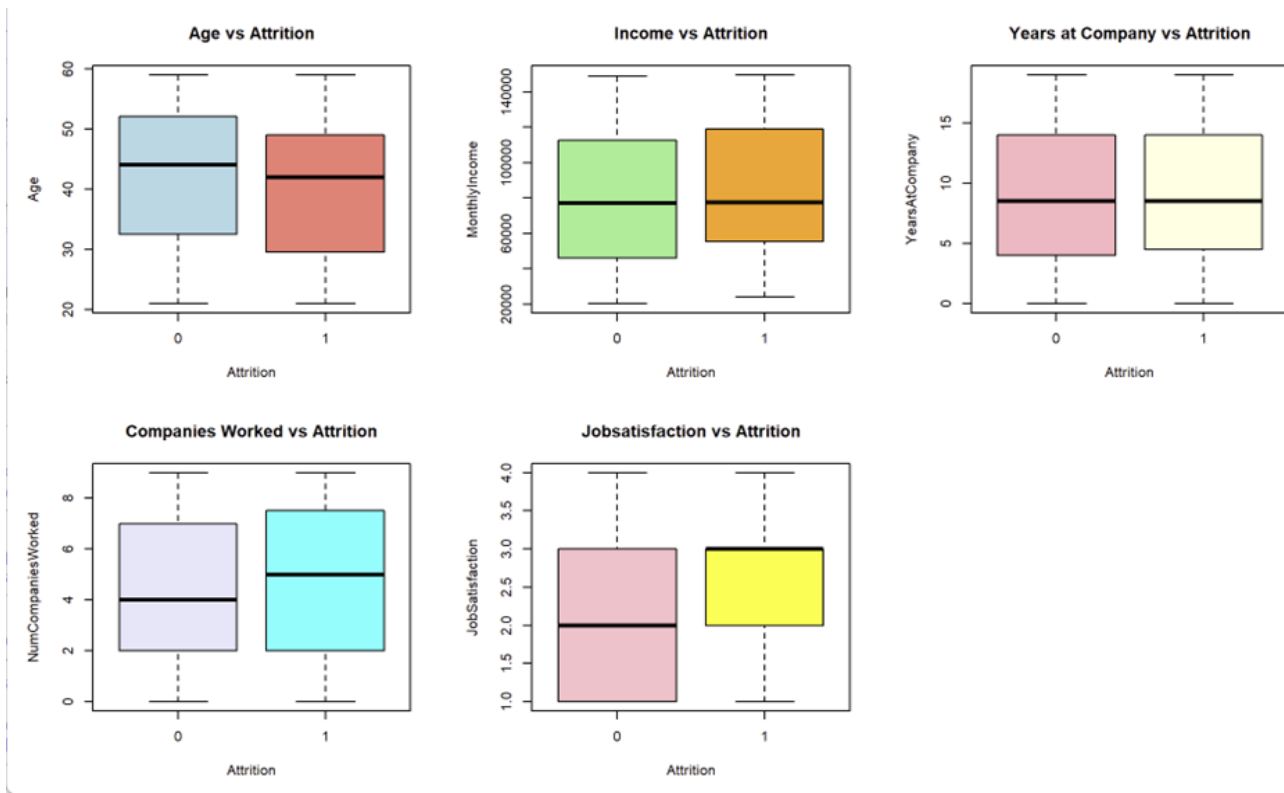
[3]

(v) Boxplot

```
par(mfrow = c(2,3))
```

```
boxplot(Age ~ Attrition, data = data,
```

```
main = "Age vs Attrition",  
  
col = c("lightblue","salmon"))  
  
boxplot(MonthlyIncome ~ Attrition, data = data,  
  
main = "Income vs Attrition",  
  
col = c("lightgreen","orange"))  
  
boxplot(YearsAtCompany ~ Attrition, data = data,  
  
main = "Years at Company vs Attrition",  
  
col = c("lightpink","lightyellow"))  
  
boxplot(NumCompaniesWorked ~ Attrition, data = data,  
  
main = "Companies Worked vs Attrition",  
  
col = c("lavender","cyan"))  
  
boxplot(JobSatisfaction ~ Attrition, data = data,  
  
main = "Jobsatisfaction vs Attrition",  
  
col = c("pink","yellow"))  
  
Output: # Boxplots for visualization
```



[5]

(vi) Interpretation**Interpretation & Findings**

- **Age:** Employees with attrition tend to be younger than those who stay, indicating higher turnover among early-career individuals. Younger employees are **more likely to leave**.
- **Monthly Income:** Median income for both groups is **very similar**, but slightly **higher for Attrition = 1**. Employees who stays tend to have **marginally lower income**, but the difference is **not very strong**. Distributions overlap heavily. Salary has **some impact**, but not very strong.

```
(> median(dataMonthlyIncome\[dataAttrition == 0], na.rm = TRUE)
```

```
[1] 77108
```

```
> median(dataMonthlyIncome\[dataAttrition == 1], na.rm = TRUE)
```

```
[1] 77624.5)
```

- **Years at Company:** Median years at Company are similar for Attrition = 1 and 0. Distributions overlap heavily. Years at Company has no impact.

```
> median(dataYearsAtCompany\[dataAttrition == 0], na.rm = TRUE)
```

```
[1] 8.5
```

```
> median(dataYearsAtCompany\[dataAttrition == 1], na.rm = TRUE)
```

```
[1] 8.5
```

- **NumCompaniesWorked:** Median is **higher for Attrition = 1**. Employees who have worked at **more companies are more likely to leave**, suggesting job-hopping behavior. Employees who left have **higher median number of companies worked**. Indicates more job switching behavior. Employees with frequent job changes are **more prone to attrition**.
- **Job Satisfaction:** Employees who left have **higher job satisfaction**. Those who stayed show **lower median satisfaction**. Median satisfaction is **lower for Attrition = 0** (this is interesting). Attrition = 1 shows **slightly higher median**, Job satisfaction alone does **not clearly explain attrition**, as medians are counterintuitive.

Conclusion

- Younger employees show higher attrition → early-career turnover is high
- Monthly income and years at company show minimal impact (similar medians)
- Employees with more prior companies are more likely to leave → job-hopping trend
- Job satisfaction does not clearly explain attrition (counterintuitive pattern which is unexpected)
- Overall, attrition is driven more by age and past mobility than salary or tenure

[3]

(vii)**Decision Tree Code**

```

library(rpart)

library(dplyr)

library(ggplot2)

library(rpart.plot)

library(scales)

# Build decision tree

tree_model <- rpart(Attrition ~ OverTime + YearsAtCompany + JobSatisfaction,

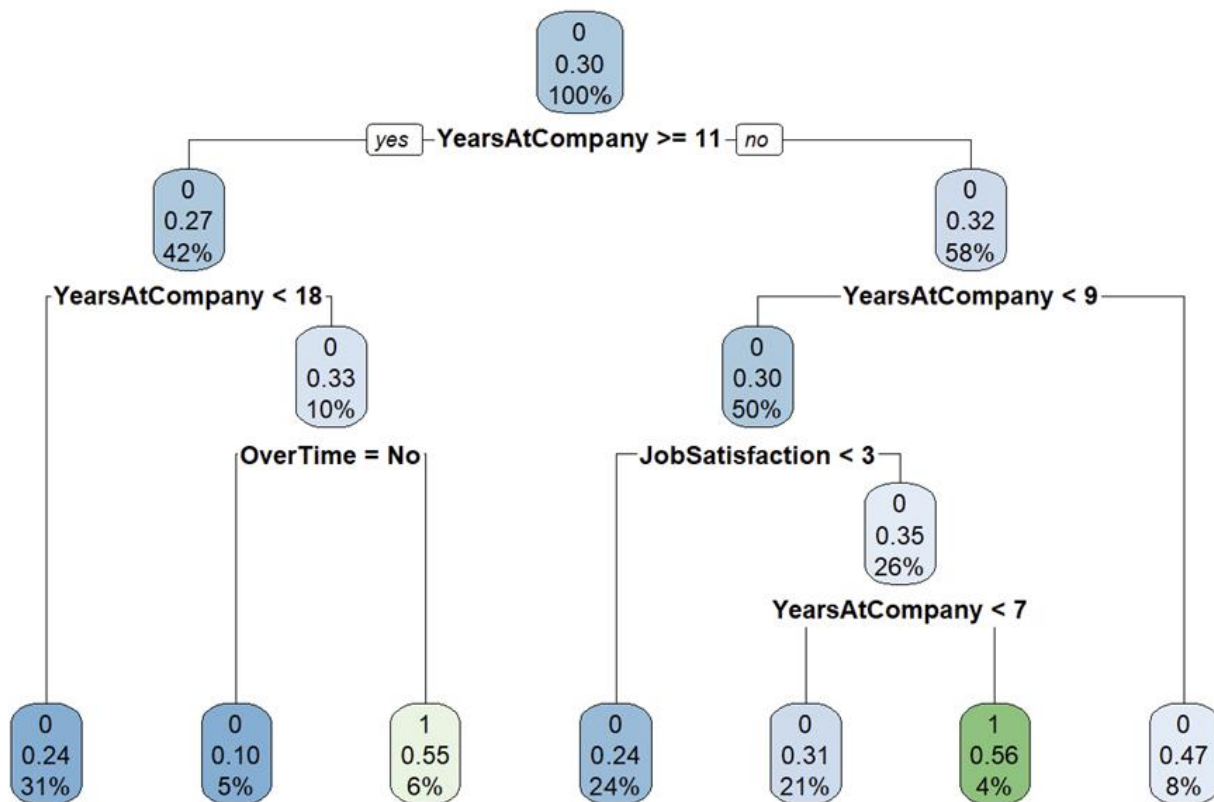
data = data,

method = "class", control = rpart.control(cp = 0.002))

# Plot tree

rpart.plot(tree_model)

```

Diagram:

Inference:

- **YearsAtCompany is the primary split**, indicating it is the strongest predictor of attrition.
- Employees with **≥ 11 years** of tenure have **lower attrition (~27%)**, showing higher stability among long-tenured staff.
 - For this group, further splits (like OverTime = No) show **very low attrition (~10%)**, confirming strong retention. While those doing overtime has 55% attrition which is obvious. Meaning overworked employees will leave company.
- Employees with **< 11 years** have **higher attrition (~32%)**, showing lower stability, especially those with **< 9 years at Company**.
 - Within this segment, **JobSatisfaction becomes critical**:
 - **High job satisfaction (3 or 4)** increases attrition risk which is 35% as explained by box plot above as well. Data explains so which is counterintuitive.
 - **Low job satisfaction has higher retention where attrition is 24% only.**
 - When combined with **very low tenure (> 7 years)**, attrition reaches the **highest level (~56%)**.
- Employees with **moderate tenure (less than 7 years)** or better satisfaction show **reduced attrition (~31%)**, indicating partial stability.

Bottom of Form

[8]

(viii)

Predicted probabilities (prob of Attrition = 1)

```
data$prob <- predict(tree_model, type = "prob")[,2]
```

	Attrition	Age	Department	MonthlyIncome	YearsAtCompany	JobSatisfaction	OverTime	NumCompaniesWorked	EducationField	prob
1	1	47	Tech	125570	6	3	No	9	HR	0.1481481
2	0	37	HR	124293	17	2	Yes	0	Finance	0.1315789
3	0	25	HR	125185	10	3	No	9	Finance	0.5
4	1	24	Tech	127825	12	1	No	6	Marketing	0.1315789
5	1	45	Tech	123196	5	4	Yes	7	Marketing	0.1315789
6	1	25	HR	124573	10	1	No	5	Marketing	0.1315789
7	1	56	Tech	125654	3	3	Yes	7	HR	0.5
8	1	53	Tech	123081	6	1	No	1	Engineering	0.2
9	1	53	HR	124680	14	3	No	5	HR	0
10	1	45	Tech	78141	7	3	Yes	6	Finance	0.6
11	1	38	Tech	76570	2	1	Yes	5	HR	0.3125
12	0	50	Tech	70612	3	4	Yes	9	Marketing	0.6
13	0	23	Tech	65106	1	3	Yes	8	Engineering	0.4705882
14	0	59	HR	78017	5	4	Yes	6	Engineering	0.5
15	1	34	Sales	57626	7	1	Yes	0	Engineering	0.1481481
16	1	59	HR	75479	4	4	Yes	1	Finance	0.2580645
17	1	36	HR	65002	2	2	Yes	1	Marketing	0.1315789
18	0	39	HR	61240	0	4	Yes	9	HR	0.5833333
19	1	32	Sales	71745	3	3	Yes	3	Finance	0.4705882
20	0	53	HR	47712	1	4	No	3	Engineering	0.1481481

[2]

(ix)

Case 1: Cutoff = 0.4

```
datapred_04 <- ifelse(dataprob > 0.4, 1, 0)
```

```
datapred_04 <- -factor(datapred_04)
```

Case 2: Cutoff = 0.6

```
datapred_06 <- -ifelse(dataprob > 0.6, 1, 0)
```

```
datapred_06 <- -factor(datapred_06)
```

Output:

	Attrition	Age	Department	MonthlyIncome	YearsAtCompany	JobSatisfaction	OverTime	NumCompaniesWorked	EducationField	prob	pred_04	pred_06
1	1	47	Tech	125570	6	3	No	9	HR	0.1481481	0	0
2	0	37	HR	124293	17	2	Yes	0	Finance	0.1315789	0	0
3	0	25	HR	125185	10	3	No	9	Finance	0.5	1	0
4	1	24	Tech	127825	12	1	No	6	Marketing	0.1315789	0	0
5	1	45	Tech	123196	5	4	Yes	7	Marketing	0.1315789	0	0
6	1	25	HR	124573	10	1	No	5	Marketing	0.1315789	0	0
7	1	56	Tech	125654	3	3	Yes	7	HR	0.5	1	0
8	1	53	Tech	123081	6	1	No	1	Engineering	0.2	0	0
9	1	53	HR	124680	14	3	No	5	HR	0	0	0
10	1	45	Tech	78141	7	3	Yes	6	Finance	0.6	1	0
11	1	38	Tech	76570	2	1	Yes	5	HR	0.3125	0	0
12	0	50	Tech	70612	3	4	Yes	9	Marketing	0.6	1	0
13	0	23	Tech	65106	1	3	Yes	8	Engineering	0.4705882	1	0
14	0	59	HR	78017	5	4	Yes	6	Engineering	0.5	1	0
15	1	34	Sales	57626	7	1	Yes	0	Engineering	0.1481481	0	0
16	1	59	HR	75479	4	4	Yes	1	Finance	0.2580645	0	0
17	1	36	HR	65002	2	2	Yes	1	Marketing	0.1315789	0	0
18	0	39	HR	61240	0	4	Yes	9	HR	0.5833333	1	0
19	1	32	Sales	71745	3	3	Yes	3	Finance	0.4705882	1	0
20	0	53	HR	47712	1	4	No	3	Engineering	0.1481481	0	0

[3]

(x) # Rank employees based on predicted attrition probabilities

```
data <- data %>% arrange(desc(prob))
```

Output (Arrange in descending order)

	Attrition	Age	Department	MonthlyIncome	YearsAtCompany	JobSatisfaction	OverTime	NumCompaniesWorked	EducationField	prob	pred_04	pred_06
1	1	31	HR	76178	8	3	No	7	Engineering	0.5555556	1	0
2	1	45	Tech	78141	7	3	Yes	6	Finance	0.5555556	1	0
3	0	44	HR	24809	7	3	No	2	HR	0.5555556	1	0
4	1	47	Sales	56187	8	4	Yes	9	Engineering	0.5555556	1	0
5	1	37	Sales	144454	7	3	Yes	9	Finance	0.5555556	1	0
6	0	50	HR	56586	8	3	Yes	6	Marketing	0.5555556	1	0
7	0	30	HR	42415	7	4	Yes	9	Finance	0.5555556	1	0
8	1	22	Sales	48732	7	3	Yes	0	Engineering	0.5555556	1	0
9	0	48	HR	141465	7	3	Yes	8	Marketing	0.5555556	1	0
10	1	38	Sales	115084	19	2	Yes	5	Engineering	0.5454545	1	0
11	0	50	Tech	66540	18	2	Yes	0	Finance	0.5454545	1	0
12	1	46	Tech	27239	18	4	Yes	5	Marketing	0.5454545	1	0
13	1	40	Sales	93963	18	1	Yes	2	Finance	0.5454545	1	0
14	0	52	Tech	64238	18	2	Yes	9	HR	0.5454545	1	0
15	0	43	Sales	21252	19	3	Yes	6	Finance	0.5454545	1	0
16	1	46	Sales	62318	19	2	Yes	9	Marketing	0.5454545	1	0
17	1	41	Tech	37144	18	1	Yes	3	Finance	0.5454545	1	0
18	0	51	Tech	145827	18	3	Yes	8	HR	0.5454545	1	0
19	1	40	Sales	77108	18	2	Yes	9	Marketing	0.5454545	1	0
20	0	49	HR	38777	19	1	Yes	8	Engineering	0.5454545	1	0

[2]

(xi) Create Deciles (10 Equal Groups)

```
data$decile <- ntile(-data$prob, 10)
```

	Attrition	Age	Department	MonthlyIncome	YearsAtCompany	JobSatisfaction	OverTime	NumCompaniesWorked	EducationField	prob	pred_04	pred_06	decile
1	1	31	HR	76178	8	3	No	7	Engineering	0.5555556	1	0	1
2	1	45	Tech	78141	7	3	Yes	6	Finance	0.5555556	1	0	1
3	0	44	HR	24809	7	3	No	2	HR	0.5555556	1	0	1
4	1	47	Sales	56187	8	4	Yes	9	Engineering	0.5555556	1	0	1
5	1	37	Sales	144454	7	3	Yes	9	Finance	0.5555556	1	0	1
6	0	50	HR	56586	8	3	Yes	6	Marketing	0.5555556	1	0	1
7	0	30	HR	42415	7	4	Yes	9	Finance	0.5555556	1	0	1
8	1	22	Sales	48732	7	3	Yes	0	Engineering	0.5555556	1	0	1
9	0	48	HR	141465	7	3	Yes	8	Marketing	0.5555556	1	0	1
10	1	38	Sales	115084	19	2	Yes	5	Engineering	0.5454545	1	0	1
11	0	50	Tech	66540	18	2	Yes	0	Finance	0.5454545	1	0	1
12	1	46	Tech	27239	18	4	Yes	5	Marketing	0.5454545	1	0	1
13	1	40	Sales	93963	18	1	Yes	2	Finance	0.5454545	1	0	1
14	0	52	Tech	64238	18	2	Yes	9	HR	0.5454545	1	0	1
15	0	43	Sales	21252	19	3	Yes	6	Finance	0.5454545	1	0	1
16	1	46	Sales	62318	19	2	Yes	9	Marketing	0.5454545	1	0	1
17	1	41	Tech	37144	18	1	Yes	3	Finance	0.5454545	1	0	1
18	0	51	Tech	145827	18	3	Yes	8	HR	0.5454545	1	0	1
19	1	40	Sales	77108	18	2	Yes	9	Marketing	0.5454545	1	0	1
20	0	49	HR	38777	19	1	Yes	8	Engineering	0.5454545	1	0	1

[2]

(xii) Calculate Attrition Rate per Decile

```
data$Attrition <- as.numeric(as.character(data$Attrition))
```

```
decile_summary <- data %>%
```

```
group_by(decile) %>%
```

```

summarise(

count = n(),

attrition_rate = mean(Attrition)

)

print(decile_summary)

# A tibble: 10 × 3
  decile count attrition_rate
  <int> <int>         <dbl>
1     1     20           0.65
2     2     20           0.55
3     3     20           0.5
4     4     20           0.25
5     5     20           0.25
6     6     20           0.3
7     7     20           0.15
8     8     20           0.15
9     9     20           0.2
10    10     20           0

```

[3]

(xiii)

Plot Deciles

```

library(ggplot2)

# Plot

ggplot(decile_summary, aes(x = decile, y = attrition_rate)) +

geom_line() +

geom_point() +

labs(

title = "Decile vs Attrition Rate",

x = "Decile",

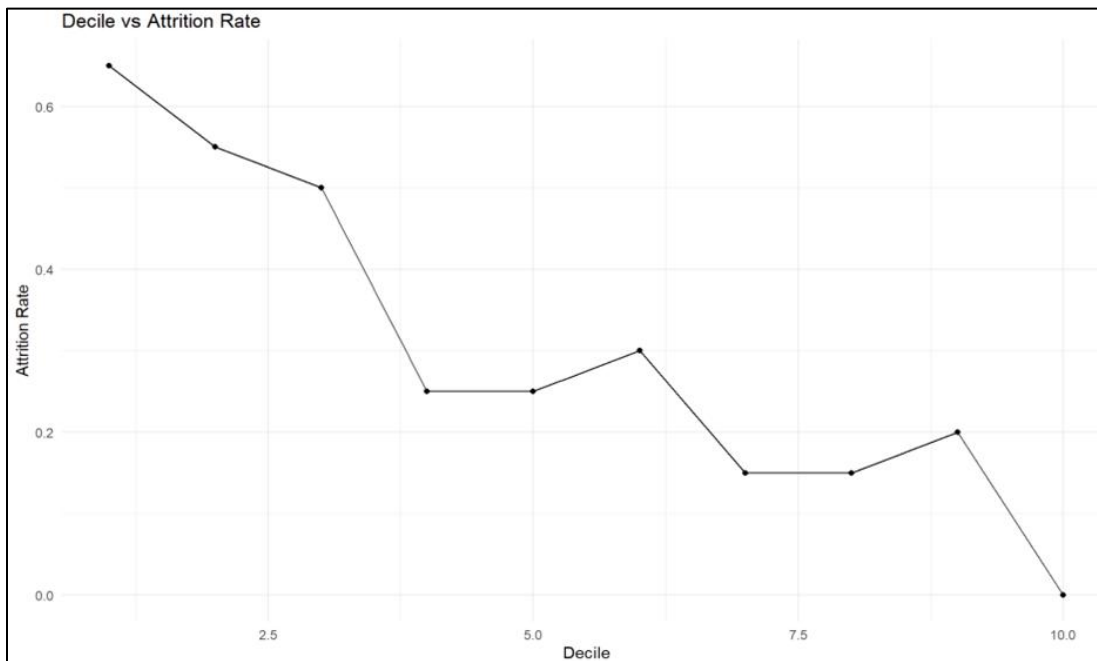
y = "Attrition Rate"

) +

theme_minimal()

```

This will plot the graph as shown below:



X-axis = Deciles

Y-axis = Attrition Rate

Inference:

- **Decile 1 → High attrition (~65%)**
- **Decile 10 → Very low attrition (~0%)**

This means:

Low deciles have higher attrition, and high deciles have lower attrition

[3]

[41 Marks]

Solution.2

(i) Uploading data in R

```
library(dplyr)
```

```
data2 <- read.csv("C:\\Users\\dipik\\OneDrive\\Desktop\\CS2B\\hospital_stay.csv")
```

```
str(data2)
```

```
# KM for entire dataset
```

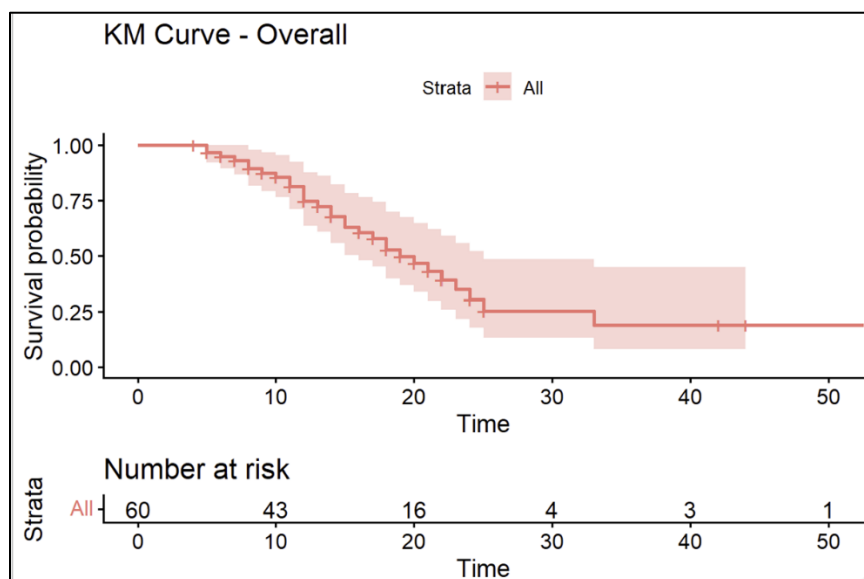
```
library(survival)
```

```
library(survminer)
```

```
km_all <- survfit(Surv(time, event) ~ 1, data = data2)
```

```
ggsurvplot(km_all, title = "KM Curve - Overall", risk.table = TRUE)
```

Output:



[3]

(ii)

KM by treatment group

```
km_group <- survfit(Surv(time, event) ~ treatment_group, data = data2)
```

```
ggsurvplot(km_group,
```

```
data = data2,
```

```
pval = TRUE,
```

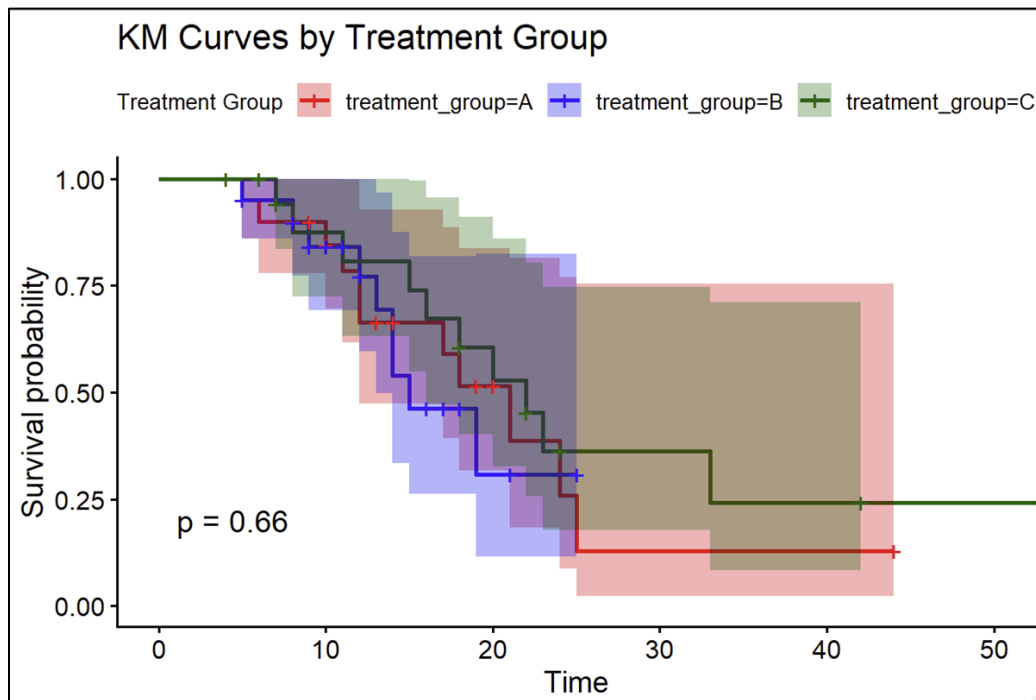
```
conf.int = TRUE,
```

```
legend.title = "Treatment Group",
```

```
palette = c("red", "blue", "darkgreen"),
```

```
title = "KM Curves by Treatment Group")
```

Output:



[3]

(iii) Log-Rank Test [4 Marks]

Objective:

Test whether survival curves differ statistically and significantly across groups.

R Code

```
survdif(Surv(time, event) ~ treatment_group, data = data2)
```

R output:

	N	Observed	Expected	(O-E) ² /E	(O-E) ² /V
treatment_group=A	20	11	10.28	0.0507	0.0788
treatment_group=B	20	9	7.48	0.3084	0.4409
treatment_group=C	20	11	13.24	0.3792	0.7426

Chisq= 0.8 on 2 degrees of freedom, p= 0.7

Look at **p-value**

- $p < 0.05 \rightarrow$ reject $H_0 \rightarrow$ treatments differ significantly
- $p \geq 0.05 \rightarrow$ no strong evidence of difference

Hence there is no strong evidence of difference across groups.

[3]

(iv) Proportional Hazards (PH) Assumption**R code:**

```
cox_model <- coxph(Surv(time, event) ~ treatment_group + age + comorbidity_score + icu_admission, data=data2)
```

```
cox.zph(cox_model)
```

R output:

	chisq	df	p
treatment_group	1.78318104	2	0.41
age	0.068564228	1	0.79
comorbidity_score	0.074806814	1	0.78
icu_admission	2.16E-09	1	1
GLOBAL	2.278430513	5	0.81

Interpretation

- $p > 0.05 \rightarrow$ PH assumption holds
- $p < 0.05 \rightarrow$ violation

“Test shows no significant time dependence ($p > 0.05$), indicating PH assumption holds.”

[4]

(v) Cox Model with Interaction

R code:

```
cox_interact <- coxph(Surv(time, event) ~ treatment_group * icu_admission + age + comorbidity_score, data=data2)
```

```
summary(cox_interact)
```

R output:

```
coxph(formula = Surv(time, event) ~ treatment_group * icu_admission +
```

```
age + comorbidity_score, data = data2)
```

n= 60, number of events= 31

coef	exp(coef)	se(coef)	
treatment_groupB	-0.32976	0.71910	19681.13503
treatment_groupC	0.06205	1.06401	19780.40500
icu_admission	25.62322	134283239515.35371	14295.62583
age	-0.06904	0.93329	0.08334
comorbidity_score	-0.82981	0.43613	0.48196
treatment_groupB:icu_admission	-0.54965	0.57715	19681.13504
treatment_groupC:icu_admission	-1.06083	0.34617	19780.40501

z Pr(>|z|)

treatment_groupB 0.000 1.0000

treatment_groupC 0.000 1.0000

icu_admission 0.002 0.9986

age -0.828 0.4074

comorbidity_score -1.722 0.0851 .

treatment_groupB:icu_admission 0.000 1.0000

treatment_groupC:icu_admission 0.000 1.0000

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

exp(coef) exp(-coef) lower .95 upper .95

treatment_groupB 0.7191 1.390633008477389 0.0000 Inf

treatment_groupC 1.0640 0.939838580035507 0.0000 Inf

icu_admission 134283239515.3537 0.000000000007447 0.0000 Inf

age 0.9333 1.071481925584763 0.7926 1.099

comorbidity_score 0.4361 2.292888521403180 0.1696 1.122

treatment_groupB:icu_admission 0.5772 1.732651224022200 0.0000 Inf

treatment_groupC:icu_admission 0.3462 2.888755655042414 0.0000 Inf

Concordance= 0.844 (se = 0.032)

Likelihood ratio test= 50.95 on 7 df, p=0.000000009

Wald test = 11.6 on 7 df, p=0.1

Score (logrank) test = 41.31 on 7 df, p=0.0000007

Interpretation:

Interpretation of Interaction

- Interaction term = treatment_group × icu_admission

Meaning:

- Effect of treatment **group (B, C vs A)** depends on ICU admission status

We can see that there are:

v **Huge standard errors (~19000)**

v **p-values = 1.000**

v **Confidence intervals = $(0, \infty)$**

This clearly indicates **Model instability**

Hence, to sum up:

v Interaction effects are **NOT statistically significant**

v Non-significant p-values

v Infinite confidence intervals

v Extremely large standard errors indicate:

- **Model is unstable**
- Results are **not reliable**

[5]

(vi) Limitations of Cox Model

1. Proportional Hazards Assumption

- The model assumes that the **hazard ratios are constant over time**
- In reality, treatment effects or risk factors may **change over time**

Ø **Limitation:**

If this assumption is violated, results become **biased or misleading**

2. Cannot Handle Time-Varying Effects Easily

- Many medical variables (e.g., blood pressure, treatment dosage) **change over time**

Ø **Limitation:**

Basic Cox model does not naturally capture **time-dependent covariates** without modification

3. Sensitive to Omitted Variables

- Missing important predictors (e.g., genetic factors, lifestyle) can affect results

Ø **Limitation:**

Leads to **confounding bias**, reducing model reliability

4. Assumes Linear Relationship (on log hazard scale)

- Continuous variables are assumed to have a **linear effect**

Ø **Limitation:**

- Real-world relationships may be **non-linear**
- Model may **oversimplify medical reality**

5. Issues with Small Sample Size

- Medical datasets often have limited patients

Ø Limitation:

- Estimates become **unstable**
- Confidence intervals become **wide**

6. Cannot Directly Estimate Survival Time

- It models **hazard (risk)**, not exact survival time

Ø Limitation:

- Less intuitive for clinicians who want **time predictions**

Final Summary

Ø The Cox model is powerful but limited by:

- **Proportional hazards assumption**
- **Inability to easily model time-varying effects**
- **Sensitivity to missing variables and small samples**

[3]

[21 Marks]

Solution.3

R Code Solutions and Outputs

(i) 95% Confidence Intervals

Formula:

$$\text{estimate} \pm 1.96 \times SE$$

R Code

```
coef_est = fitcoefse = sqrt(diag(fitvar.coef))
```

```
lower = coef_est - 1.96*se
```

```
upper = coef_est + 1.96*se
```

```
cbind(lower, upper)
```

R Output

	lower	upper
ar1	-0.2309304	0.7251
ma1	-0.4799648	0.5194
intercept	-0.1560816	0.1103

Values	
coef_est	Named num [1:3] 0.2471 0.0197 -0.0229
lower	Named num [1:3] -0.231 -0.48 -0.156
se	Named num [1:3] 0.244 0.255 0.068
upper	Named num [1:3] 0.725 0.519 0.11

[4]

(ii) Five-step forecasts

```
library(forecast)
fc = forecast(fit, h=5)
fc$mean
```

R Output

```
Time Series:
Start = 361
End = 365
Frequency = 1
[1] 0.13880460 0.01706371 -0.01301897 -0.02045252 -0.02228938
```

Name	Type	Value
fc	list [10] (S3: forecast)	List of length 10
method	character [1]	'ARIMA(1,0,1) with non-zero mean'
model	list [15] (S3: Arima)	List of length 15
level	double [2]	80 95
mean	double [5] (S3: ts)	0.1388 0.0171 -0.0130 -0.0205 -0.0223
lower	double [5 x 2] (S3: mts, ts, matrix)	-1.08 -1.25 -1.28 -1.29 -1.29 -1.73 -1.92 -1.95 -1.96 -1.96 ...
upper	double [5 x 2] (S3: mts, ts, matrix)	1.36 1.28 1.25 1.25 1.24 2.01 1.95 1.92 1.92 1.91 ...
x	double [360] (S3: ts)	-1.270 -0.786 -1.285 -1.067 -0.280 -1.508 ...
series	character [1]	'x'
fitted	double [360] (S3: ts)	-0.0676 -0.3539 -0.2200 -0.3558 -0.2949 -0.0861 ...
residuals	double [360] (S3: ts)	-1.2021 -0.4321 -1.0652 -0.7109 0.0148 -1.4220 ...

[3]

(iii) Matrix B

```
se <- (fcupper[, 2] - fcmean) / 1.96

B <- cbind(

mean = fc$mean,

se = se,

upper_95 = fc$upper[,2]

)

B
```

R Output:

	mean	se	upper_95
1	0.13880460	0.9528412	2.006373
2	0.01706371	0.9861806	1.949978
3	-0.01301897	0.9881799	1.923814
4	-0.02045252	0.9883019	1.916619
5	-0.02228938	0.9883093	1.914797

[4]

(iv) Forecast plot

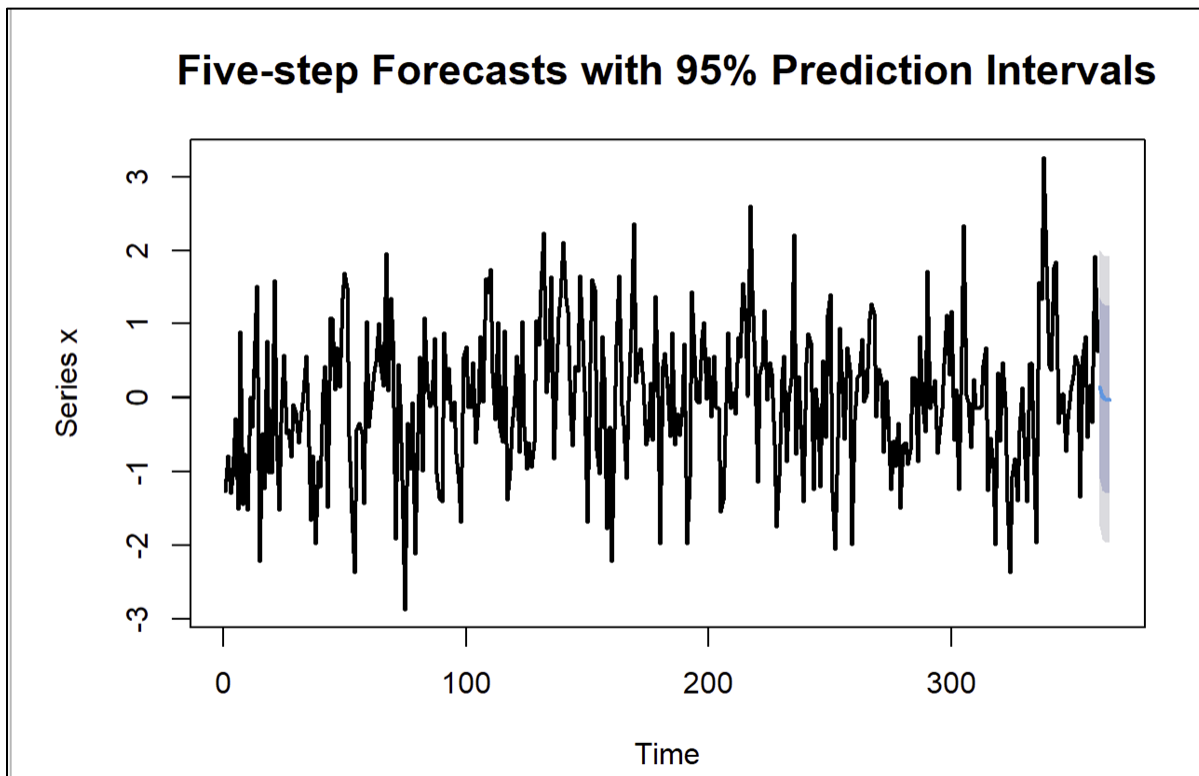
```
plot(fc,

main="Five-step Forecasts with 95% Prediction Intervals",

ylab="Series x",

xlab="Time")

lines(x)
```



This produces a plot with

- historical data
- forecasts
- prediction intervals.

[3]

[14 Marks]

Solution.4

(i) Construct Transition Matrix

Method

From observed transitions:

$$P_{ij} = \frac{\text{Number of transitions from } i \rightarrow j}{\text{Total transitions from } i}$$

R Code:

```
# Create transition table
```

```
trans_mat <- table(current, future)
```

Working explanation:

Suppose:

```
<> R
current <- c(1,2,2,3,2)
future  <- c(2,2,3,4,3)
```

Each pair represents a transition:

Current	Future	Meaning
1	2	$1 \rightarrow 2$
2	2	$2 \rightarrow 2$
2	3	$2 \rightarrow 3$
3	4	$3 \rightarrow 4$
2	3	$2 \rightarrow 3$

R output:

```
      future
current 1 2 3 4
  1 0 2 0 0
  2 1 1 2 0
  3 0 0 0 3
  4 0 0 0 1
```

Convert to probabilities

```
P <- prop.table(trans_mat, 1)
```

P

R Output:

```
      future
current 1    2    3    4
  1 0.00 1.00 0.00 0.00
  2 0.25 0.25 0.50 0.00
  3 0.00 0.00 0.00 1.00
  4 0.00 0.00 0.00 1.00
```

[4]

(ii) Check if “Lost” is Absorbing

Condition

A state is absorbing if:

$$P_{44} = 1$$

Check P_{44} value

R Code:

```
is_absorbing <- P[4,4] == 1
```

```
is_absorbing
```

R output:

```
TRUE
```

Interpretation

- If = 1 → absorbing
- If < 1 → not absorbing

[2]

(iii) simulating a Markov Chain

```
simulate_chain <- function(P, start, steps){  
  states <- numeric(steps)  
  states[1] <- start  
  for(i in 2:steps){  
    states[i] <- sample(1:4, 1, prob = P[states[i-1], ])  
  }  
  return(states)  
}  
  
set.seed(1)  
  
sim <- simulate_chain(P, 1, 5000)
```

R Output:

```
Simulations
```

Long-run proportions

```
table(sim)/length(sim)
```

R output:

```
sim
  1      2      3      4
0.0002 0.0002 0.0002 0.9994
> |
```

How is this calculated? i.e.

New, Active, Dormant = $1/5000 = .0002$

Lost = $4997/5000 = .9994$

Eventually all customers → Lost

Matches theoretical absorption probability

Why long term probabilities of 5,000 steps clearly represent the theoretical stationary distribution of the process?

Candidates should just mention that state 4 is the absorbing state hence, long term probabilities of 5,000 steps clearly represent the theoretical stationary distribution of the process

[5]

(iv) Compute the 2-step and 3-step transition probability matrices

R code:

```
P2 <- P %*% P
P3 <- P2 %*% P
```

```
P2
P3
```

R Output:

```
> P
      future
current  1      2      3      4
  1 0.00  1.00  0.00  0.00
  2 0.25  0.25  0.50  0.00
  3 0.00  0.00  0.00  1.00
  4 0.00  0.00  0.00  1.00

> P2
      future
current  1      2      3      4
  1 0.2500 0.2500 0.500 0.0
  2 0.0625 0.3125 0.125 0.5
  3 0.0000 0.0000 0.000 1.0
  4 0.0000 0.0000 0.000 1.0

> P3
      future
current  1      2      3      4
  1 0.062500 0.312500 0.12500 0.500
  2 0.078125 0.140625 0.15625 0.625
  3 0.000000 0.000000 0.00000 1.000
  4 0.000000 0.000000 0.00000 1.000
```

[4]

(v)

Probability of being in “Lost” after k steps**R code:**

```
library(expm)

initial <- c(1,0,0,0)

P2 <- P %*% P

P5 <- P %^% 5

prob_2 <- initial %*% P2

prob_5 <- initial %*% P5

prob_2

prob_5
```

R output:

```
P5 =

> P5
      future
current 1      2      3      4
1 0.03515625 0.11328125 0.07031250 0.7812500
2 0.02832031 0.06347656 0.05664062 0.8515625
3 0.00000000 0.00000000 0.00000000 1.0000000
4 0.00000000 0.00000000 0.00000000 1.0000000

> |

prob_2

future

1 2 3 4

[1,] 0.25 0.25 0.5 0

prob_5

future

1 2 3 4

[1,] 0.03515625 0.1132812 0.0703125 0.78125
```

[4]

(vi)

Conditional Probability**R Code:**

```

P2 <- P %*% P

# Required probability

#  $P(X_{t+2} = 4 \mid X_t = 2 \text{ AND } X_{t+1} \neq 3)$ 

# Step-wise calculation

p_21 <- P[2,1]

p_22 <- P[2,2]

# paths avoiding state 3

prob <- p_21 P[1,4] + p_22 P[2,4]

prob

```

R Output

```
[1] 0
```

Interpretation

- No path from state 2 to 4 in 2 steps without passing through state 3
- Hence probability = 0

[3]**(vii) R Code:**

```

P[1,2] P[2,3] P[3,4]

1*0.5*1

```

R output:

```
[1] 0.5
```

50% chance of this exact journey

See this matrix

> P

		future			
current		1	2	3	4
1	0.00	1.00	0.00	0.00	
2	0.25	0.25	0.50	0.00	
3	0.00	0.00	0.00	1.00	
4	0.00	0.00	0.00	1.00	

[2]

[24 Marks]
