Survival Modeling of Ultradrug in Oncology

Alice Lai

Section 1: Survival analysis

1.1 Produce appropriate non-parametric summaries of the data. Does the trial data suggest a beneficial effect for the new treatment? [10 marks]

A total of 500 patients were enrolled in the trial, equally randomised between the Ultradrug and control (best supportive care) arms, with 250 participants in each group. The median age was 69 years (range 17–92), and the maximum follow-up period extended to 1096 days, approximately 36 months. Across the cohort, 58% of patients experienced a progression-free survival (PFS) event, while the remaining 42% were censored. There were no missing data.

Kaplan–Meier (KM) survival estimates revealed clear separation between treatment groups over the full 36-month period. The median PFS was 17.7 months (95% CI: 15.1-23.0) for the control group and 26.8 months (95% CI: 21.7-30.0) for the Ultradrug group, suggesting a substantial improvement in PFS. The confidence intervals showed minimal overlap, indicating that this improvement was clinically meaningful.

To further assess the treatment effect over time, restricted mean survival time (RMST) was calculated using a truncation time of 36.01 months. The RMST for the control group was 19.7 months (95% CI: 18.0-21.4), while that for the Ultradrug group was 23.5 months (95% CI: 22.0-25.1). The estimated difference in RMST was 3.8 months (95% CI: 1.5–6.1), again favouring Ultradrug and demonstrating both statistical and clinical significance.

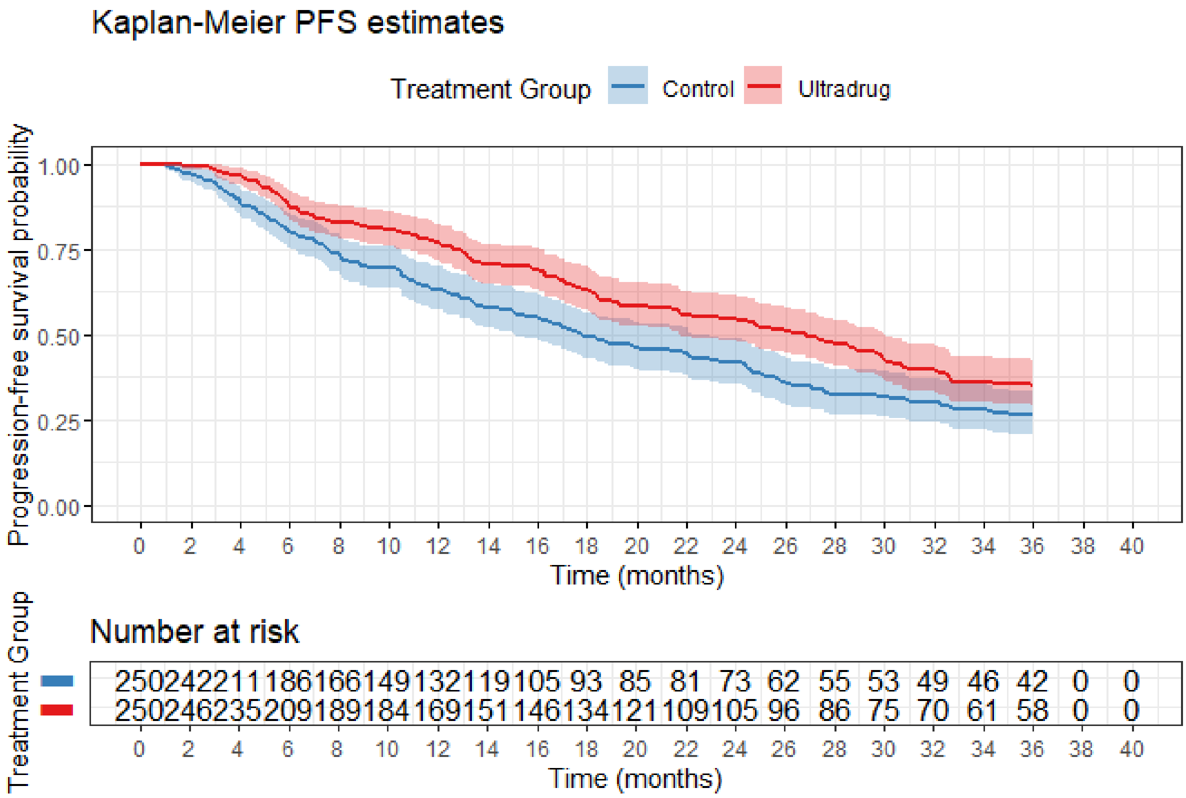
Finally, the log-rank test yielded a p-value of 0.003, providing strong statistical evidence against the null hypothesis of identical survival distributions. Taken together, the non-parametric analyses strongly suggest a beneficial effect of Ultradrug.

1.2 State and briefly describe the modelling assumption which underlies the fitting of a combined exponential model to the experimental and control groups (with treatment group as a covariate). Assess the suitability of this assumption based on the observed data. [9 marks]

The combined exponential model assumes a constant hazard rate and a proportional, time-invariant treatment effect, implemented via a shared baseline hazard and a fixed hazard ratio across treatment arms. However, this proportional hazard (PH) assumption, which requires stable relative risk over time, is not supported by diagnostic findings in this dataset.

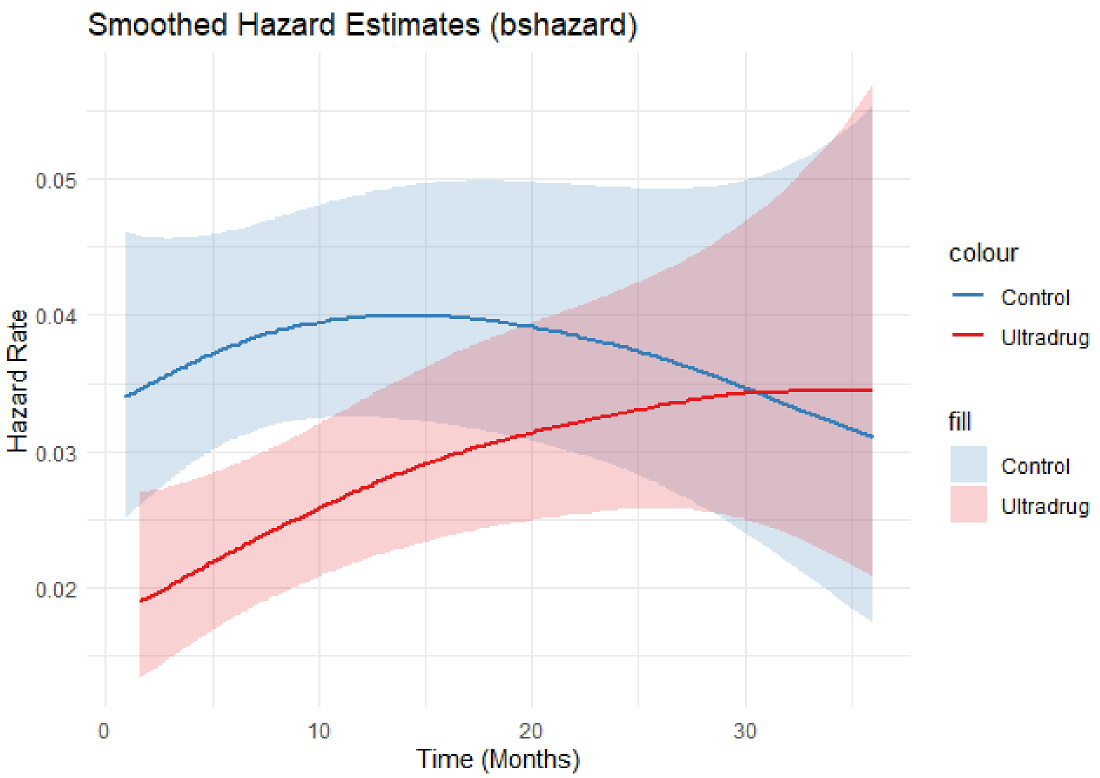
The Kaplan-Meier curves (Figure 1) display non-parallel separation between groups, particularly during the first 24 months, suggesting time-varying treatment effects.

Figure 1 Progression-Free of Ultradrug Compared to Control (KM Estimate)



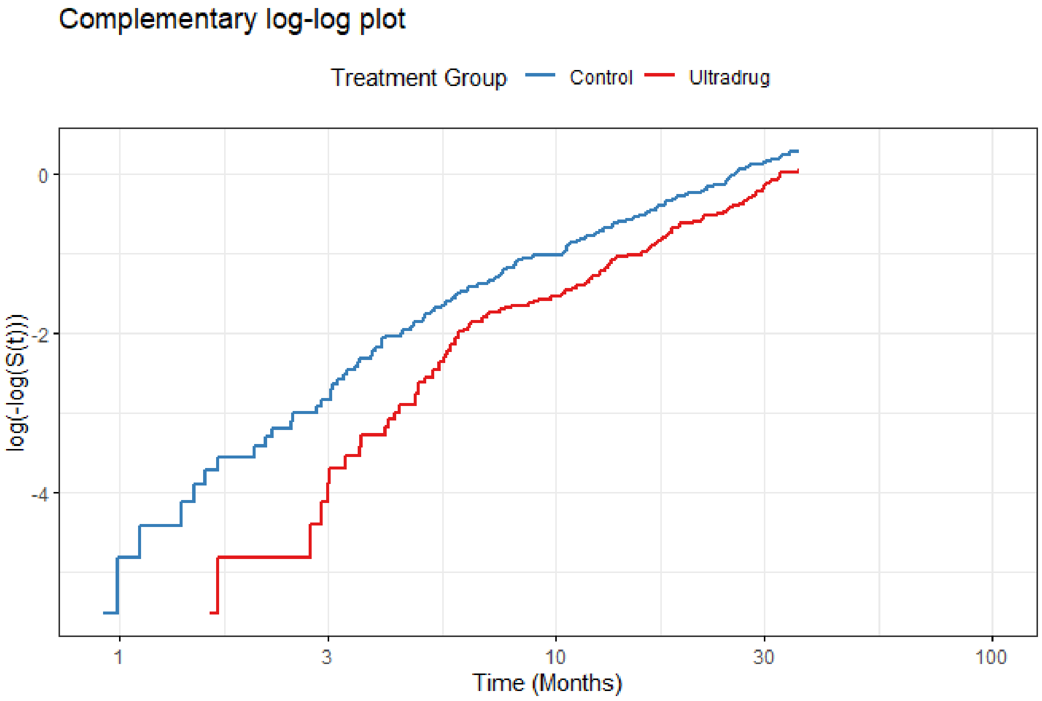
Smoothed hazard plots (Figure 2) reveal distinct hazard patterns: the control group exhibits an initial increase and subsequent decline, while the Ultradrug group shows a gradual rise that flattens out. Notably, the hazard functions converge around 30 months, which contradicts the notion of a constant hazard ratio.

Figure 2 Smoothed Hazard Functions: Ultradrug vs. Control



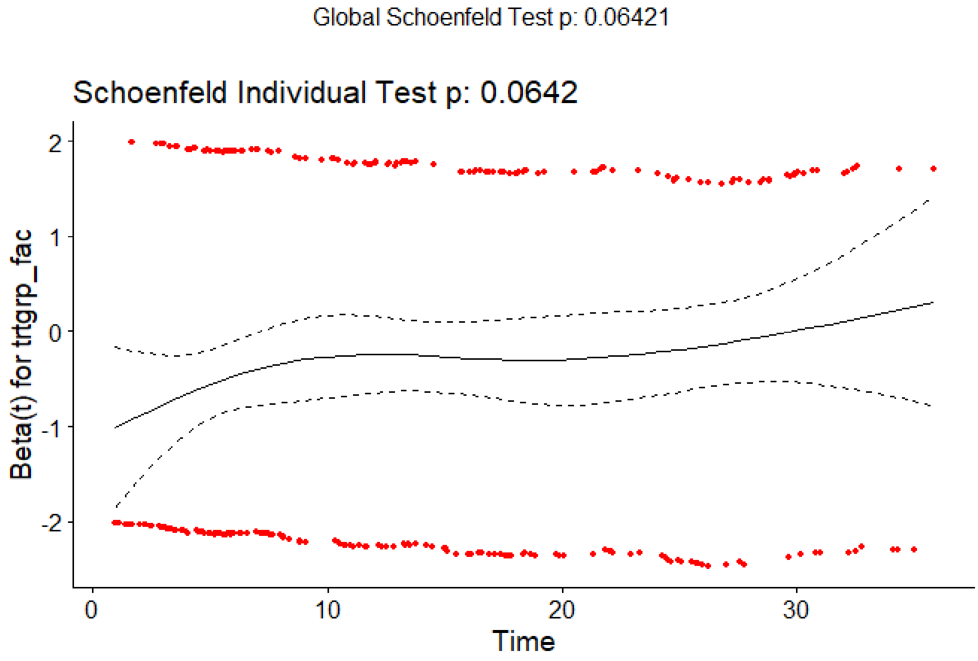
The complementary log–log plots (Figure 3) show a divergence from parallelism beyond 20 months, indicating potential violation of the PH assumption.

Figure 3 Complementary Log–Log Plot by Treatment Group



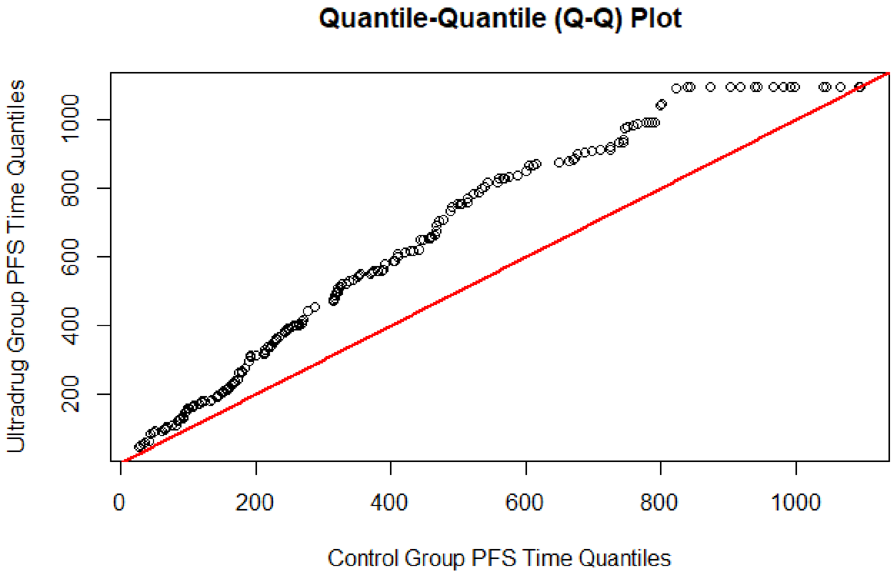
Additionally, the Schoenfeld residuals test (Figure 4) yielded a borderline p-value of 0.064. Although not conventionally significant, this result raises further concern about time-dependent effects.

Figure 4 Schoenfeld Residuals for Treatment Effect Over Time



The Q–Q plot (Figure 5) also indicates a structural divergence in survival distributions between groups.

Figure 5 Quantile-Quantile (Q-Q) Plot of PFS Times: Ultradrug vs. Control



Clinical insight supports these observations: Ultradrug is discontinued after two years, and its therapeutic effect is expected to wane thereafter. This anticipated decline is inconsistent with the assumption of a constant treatment effect, suggesting alternative models should be considered for more accurate extrapolation.

1.3 Fit the combined exponential model to the data. State and briefly interpret the estimated treatment effect. What is the mean expected PFS benefit for the new treatment vs current best supportive care based on this model? [6 marks]

A combined exponential model was fitted, with treatment group (0 = control, 1 = Ultradrug) entered as a covariate.

The fitted model produced the following estimates: the baseline hazard rate (control group) was 0.0373 per person-month. The log hazard ratio for Ultradrug versus control was −0.325 (95% CI: −0.555 to −0.094), corresponding to a hazard ratio of 0.723 (95% CI: 0.574 to 0.910). This indicates that Ultradrug reduced the risk of progression or death by approximately 28%, with statistical significance.

Using the exponential model formula for mean survival (1/λ), the estimated mean PFS was 26.78 months for the control group and 37.05 months for the Ultradrug group, resulting in a mean difference of 10.27 months.

However, this estimate exceeds the RMST difference (3.8 months) calculated earlier, along with visual diagnostics highlighting model limitations. The hazard function (Figure 6) comparison shows clear deviation from the constant hazard assumption, particularly after 25 months where convergence and crossing occur. Similarly, the modelled survival curves (Figure 7) deviate from the Kaplan-Meier estimates, underestimating early survival in the Ultradrug arm and overestimating late survival in the control arm.

Figure 6 Hazard Function: Exponential Model vs. Kaplan-Meier Estimates

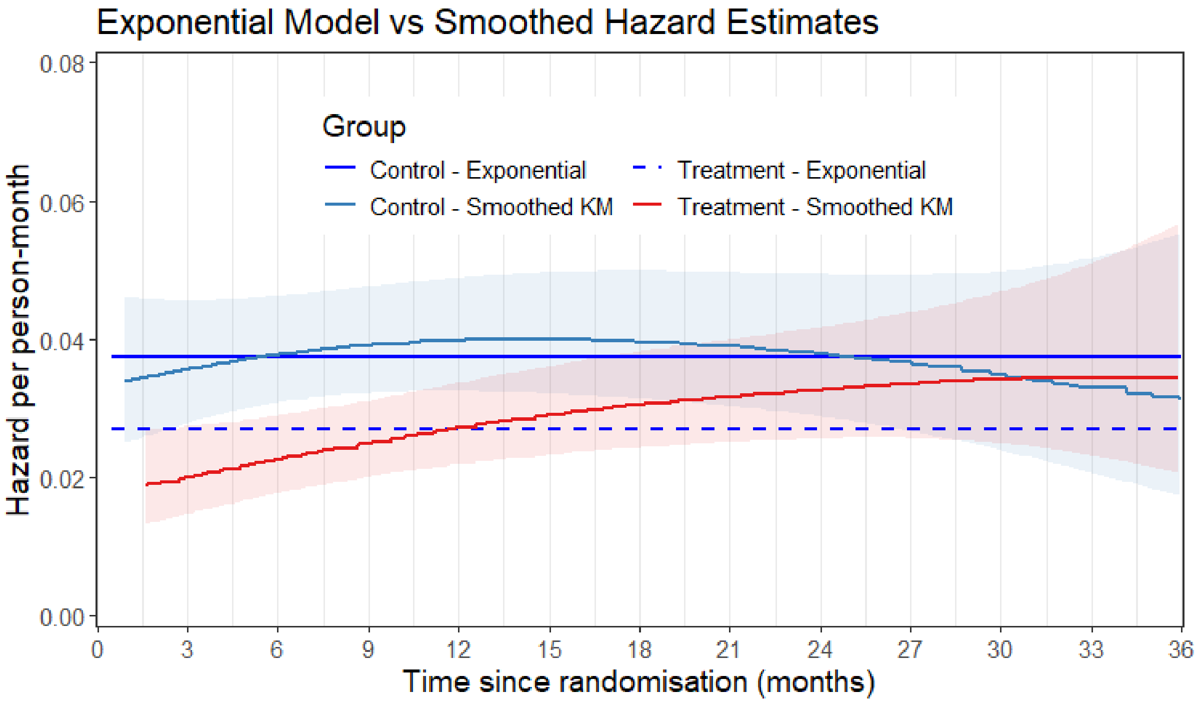
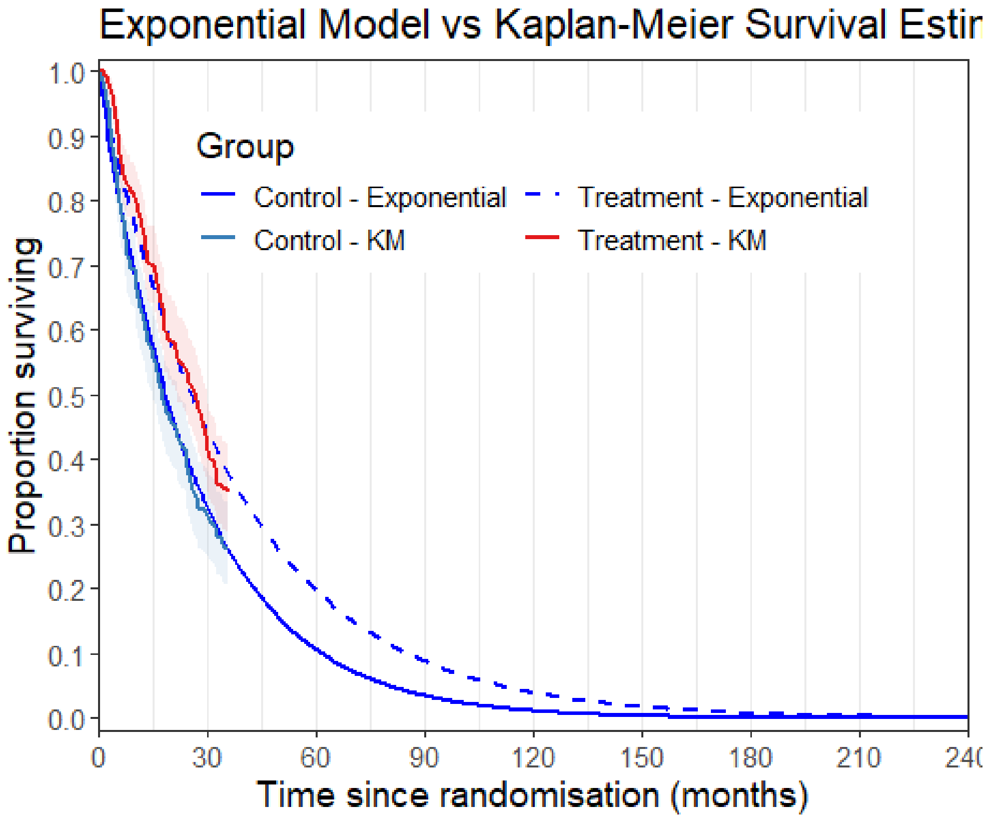


Figure 7 Survival Curves: Exponential Model vs. Kaplan-Meier Estimates



1.4 Use the statistical procedures that you have learned in HAR6178, together with the clinical advice, to present a thorough model selection that does not impose a single treatment effect for the entire time period. Make a clear recommendation on which model(s) should be considered for the economic model. [27 marks]

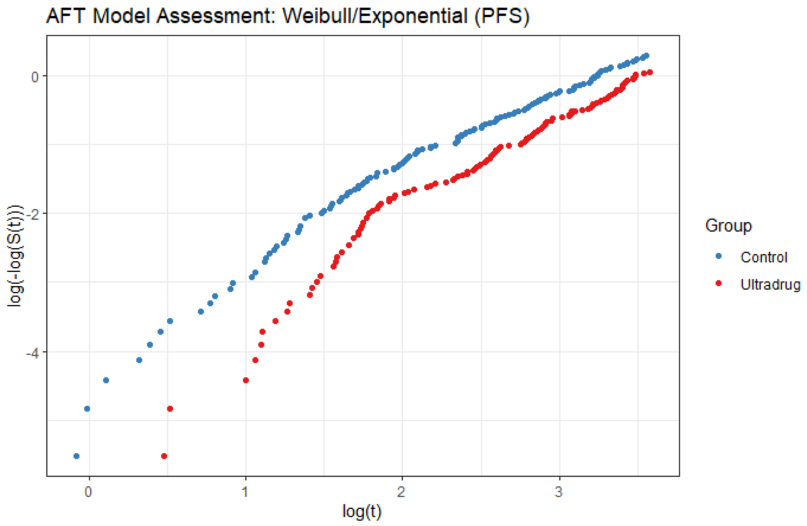
To support economic modelling with realistic progression-free survival (PFS) projections, parametric models must accommodate time-varying treatment effects, particularly waning efficacy after Ultradrug discontinuation. Model selection followed two steps: assumption checking via graphical diagnostics, and comparative model fitting under both combined and independent specifications, with final evaluation based on the latter.

Step 1: Assumption Checks for AFT Distributions

Graphical diagnostics were used to assess the compatibility of common AFT distributions with observed PFS behavior.

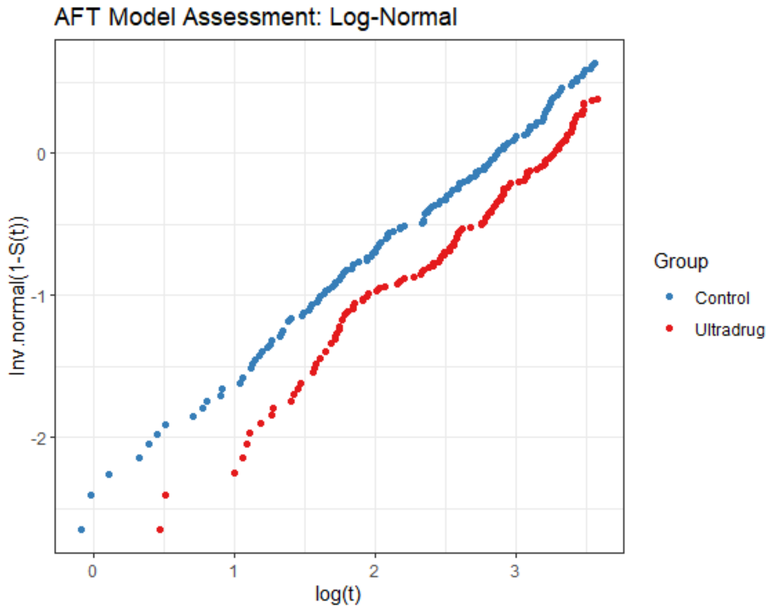
Exponential and Weibull models (Figure 8), assessed via complementary log-log plots, showed consistent deviation from linearity in both arms--especially in early and mid-follow-up--suggesting violations of constant and monotonic hazard assumptions.

Figure 8 AFT Model Diagnostic Plot: Weibull/Exponential Fit for PFS



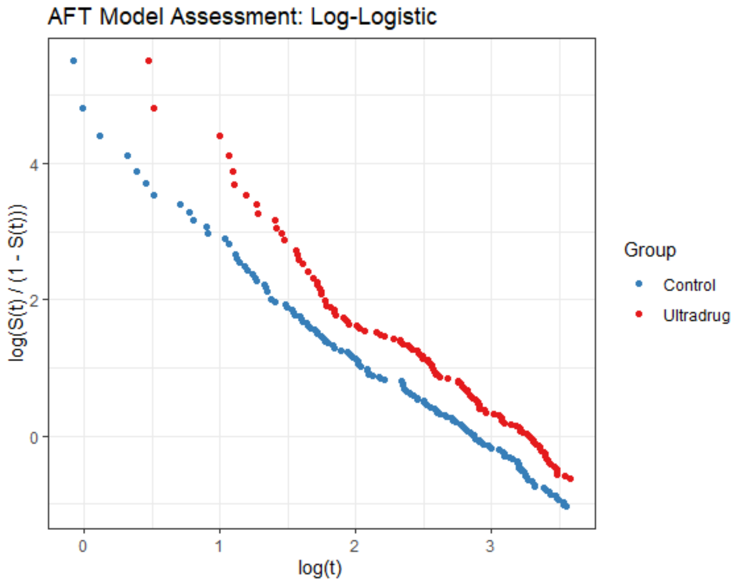
Log-normal (Figure 9), evaluated through inverse-normal plots, both groups show an approximately linear pattern, especially in the mid-to-late time periods. Although there is some deviation in the early phase (log(t) < 1.5), it appears to provide fits to captures the asymmetric and long-tailed nature of survival data.

Figure 9 AFT Model Diagnostic Plot: Log-Normal Fit



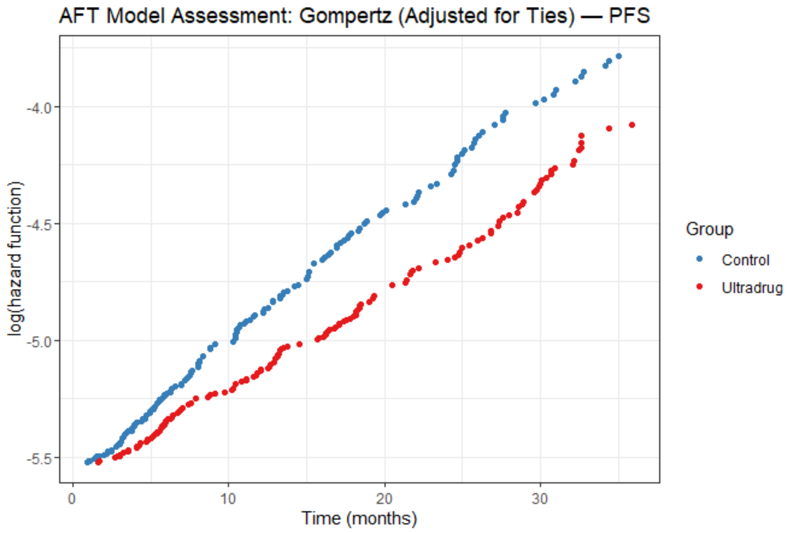
Log-logistic (Figure 10), examined via log-odds plots, both groups show a slight S-shaped curve, indicating that the log-logistic model fits certain time intervals but lacks overall linearity

Figure 10 AFT Model Diagnostic Plot: Log-Logistic Fit



Gompertz (Figure 11), evaluated using log-hazard versus time plots, exhibited linear increases consistent with rising risk. However, its structural limitation prevents capturing post-treatment hazard flattening or reversal.

Figure 11 AFT Model Diagnostic Plot: Gompertz Fit (Adjusted for Ties)



Step 2: Model Fitting and Evaluation

All models were fitted under both combined and independent specifications (Table 1). Combined models assume a shared baseline hazard and constant treatment effect, limiting their ability to reflect treatment waning or hazard convergence. Independent models, by contrast, estimate distinct hazard functions for each arm--better capturing clinical realities such as declining Ultradrug effect and delayed control-group stabilization.

Table 1 Summary of model fit statistics and treatment effect estimates.

Model	Treatment Effect (HR, Combined exp(est))	AIC (Combined)	BIC (Combined)	AIC (Control)	AIC (Ultradrug)
Exponential	0.72282	2579.845	2588.274	1314.057	1265.788
Weibull	0.70448	2568.137	2580.781	1312.936	1254.633
Log-normal	1.474	2546.947	2559.591	1300.556	1247.208
Log-logistic	1.4534	2555.503	2568.147	1305.032	1251.333
Gompertz	0.71383	2578.477	2591.121	1316.019	1261.824
Gamma	0.73976	2563.717	2576.36	1311.111	1252.497
Generalized Gamma	1.4891	2548.675	2565.533	1302.399	1249.177

Though some models (e.g., log-normal, generalised gamma) performed similarly under the combined fit, independent specifications revealed more clinically meaningful differences and were used for comparison and plotting.

The **exponential model**, while simple, imposed constant hazards and missed time-varying dynamics. Its extrapolation was unrealistic and unsuitable for primary use.

The **Weibull model** allowed increasing hazard (shape >1) but assumed monotonicity. Divergent shape parameters across arms indirectly violated proportional hazards. Its inability to capture plateauing limits its role to sensitivity analysis.

The **log-normal model** emerged as the preferred option. It estimated an acceleration factor of 1.47 (95% CI: 1.19–1.83), with median PFS of 17.3 months (control) and 24.5 months (Ultradrug). It captured early benefit, post-treatment waning, and long-term survival, aligning closely with KM curves and clinical expectations.

Log-logistic showed similar flexibility but underperformed in early fit. While interpretable, it lagged log-normal in both fit and extrapolative behavior.

Gompertz, with a low shape parameter, approximated exponential behavior. Though it reflected early hazard rise, it could not model later flattening and was excluded from base-case consideration.

Gamma resembled Weibull without offering additional clinical or statistical insight.

Generalised gamma, while structurally flexible, produced a Q estimate near zero (−0.125; 95% CI: −0.59 to 0.35), behaving similarly to log-normal. Its complexity did not yield superior fit, though it remains a valuable sensitivity analysis tool.

Model behavior and clinical plausibility.

Models like log-normal and generalised gamma appeared similar under combined specification; however, independent fits revealed clinically meaningful divergence. As a result, visual diagnostics and extrapolation focus on selected independent models based on fit, interpretability, and clinical relevance:

- Log-normal: Preferred base case. It offers strong fit, long-tailed survival, and non-monotonic hazard patterns aligned with observed PFS.
- Weibull: Included as a proportional hazards comparator, though limited by its monotonic hazard structure.
- Generalised gamma: Structurally flexible and valuable for sensitivity testing, though its behavior closely resembles log-normal here.

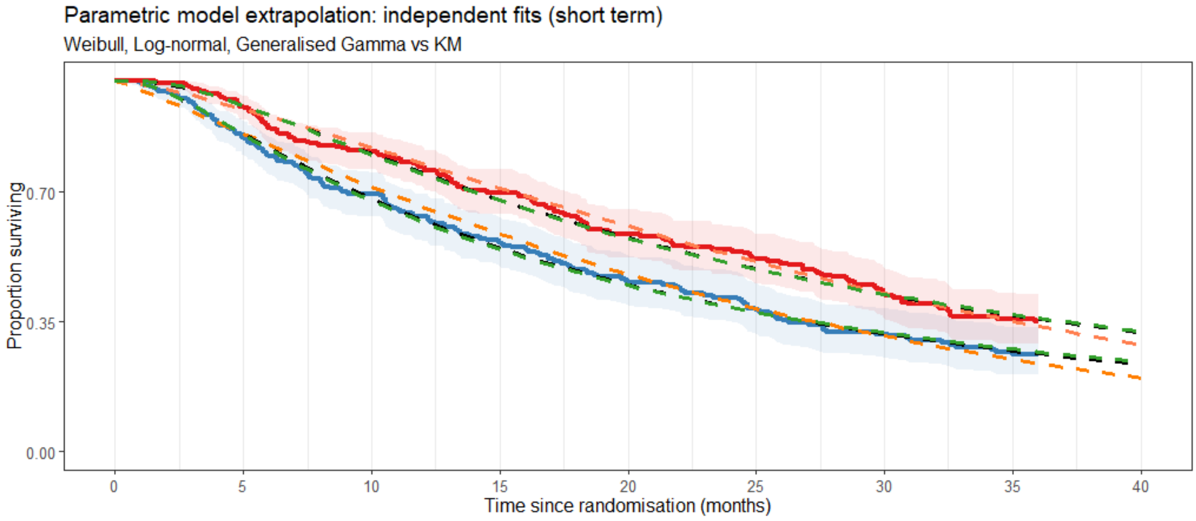
Other models (e.g., exponential, log-logistic, gamma, Gompertz) were excluded from further plots due to limited fit or clinical value, ensuring diagnostic clarity.

Visual Diagnostics and Extrapolation Behavior

Survival overlays (Figure 12) showed log-normal tracking KM estimates most closely, especially between 12-30 months. Weibull underpredicted late survival, while generalised gamma was slightly optimistic but clinically plausible.

Figure 12 Short-Term Fit Comparison of Parametric Survival Models

Weibull, Log-normal, and Generalised Gamma vs. Kaplan–Meier



Blue solid line: *Control - KM (solid)*: Kaplan–Meier survival curve for the control group

Red solid line: *Experimental - KM (solid)*: Kaplan–Meier survival curve for the Ultradrug group

Orange dashed line: Control - *Weibull (dashed)*: Weibull model fit for the control group

Coral dashed line: *Experimental - Weibull (dashed)*: Weibull model fit for the Ultradrug group

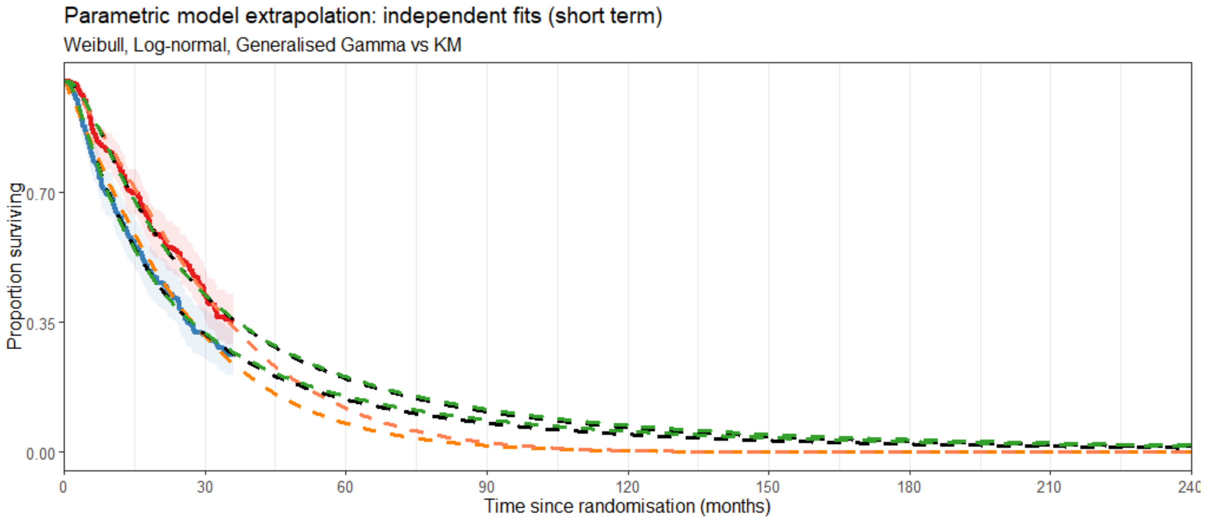
Black dashed line: *Control/Experimental - Log-normal (dashed)*: Log-normal model fits for both groups (same color used)

Green dashed line: *Control/Experimental - GenGamma (dashed)*: Generalised Gamma model fits for both groups (same color used)

Extrapolation plots (Figure 13) confirmed that log-normal and generalised gamma preserved long-term survival beyond 60 months, supporting the potential for durable benefit. Weibull declined sharply, likely underestimating QALYs.

Figure 13 Long-Term Fit Comparison of Parametric Survival Models

Weibull, Log-normal, and Generalised Gamma vs. Kaplan–Meier



Blue solid line: *Control - KM (solid)*: Kaplan–Meier survival curve for the control group

Red solid line: *Experimental - KM (solid)*: Kaplan–Meier survival curve for the Ultradrug group

Orange dashed line: Control - *Weibull (dashed)*: Weibull model fit for the control group

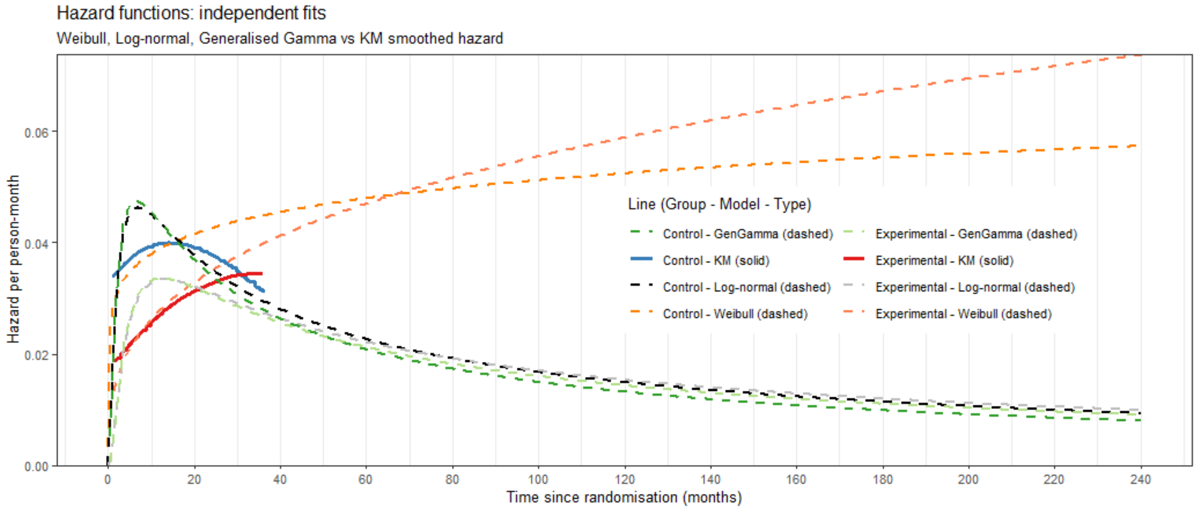
Coral dashed line: *Experimental - Weibull (dashed)*: Weibull model fit for the Ultradrug group

Black dashed line: *Control/Experimental - Log-normal (dashed)*: Log-normal model fits for both groups (same color used)

Green dashed line: *Control/Experimental - GenGamma (dashed)*: Generalised Gamma model fits for both groups (same color used)

Hazard function comparisons (Figure 14) highlighted log-normal's ability to reflect early hazard rise and post-treatment flattening. Weibull imposed continuous increases; generalised gamma offered smoother decay.

Figure 14 Hazard Function Comparison: Independent Parametric Fits vs. KM Smoothed Estimates: *Weibull, Log-normal, and Generalised Gamma Models*



Implied hazard ratios (Figure 15 & 16) further illustrated dynamic treatment effects. Log-normal showed a rising HR from ~0.2 to ~0.9, consistent with diminishing benefit. Weibull's HR exceeded 1.0 after 90 months, suggesting extrapolative artefacts.

Figure 15 Implied Hazard Ratio over Time (Short Term)

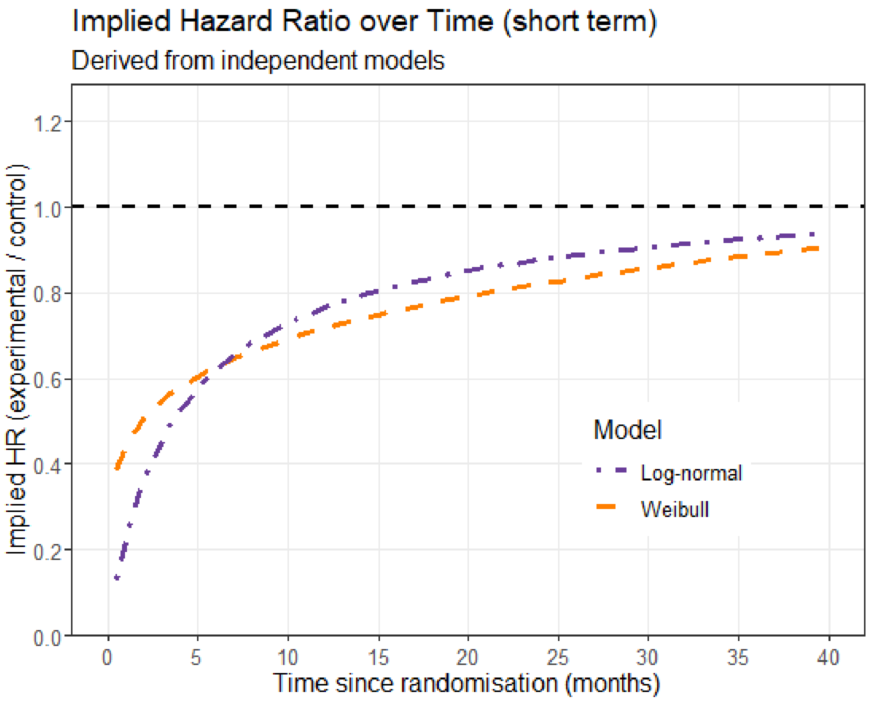
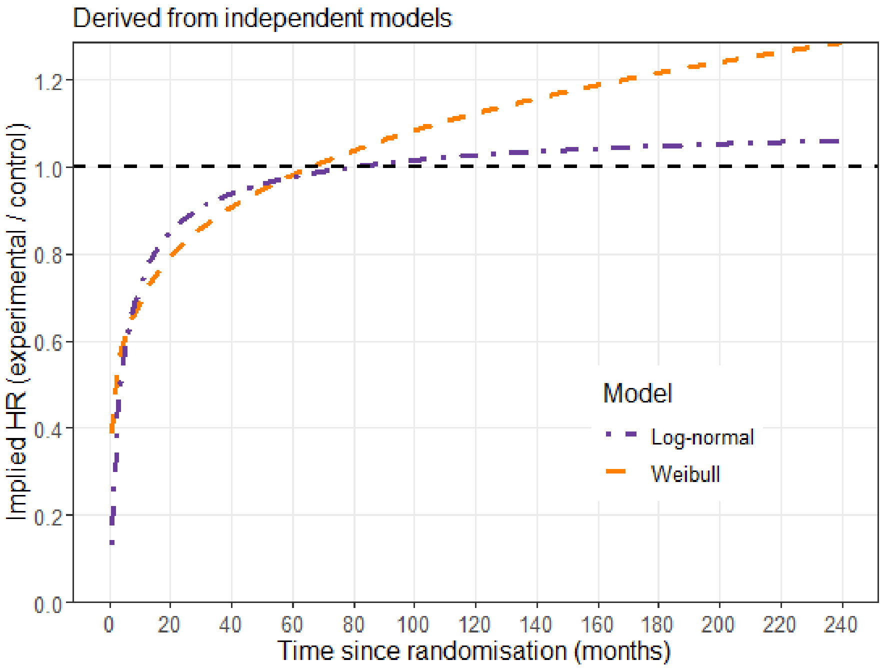


Figure 16 Implied Hazard Ratio over Time (Long Term)



Final Recommendation

The log-normal (independent) model is recommended as the base-case survival input. It fits the observed data well, aligns with clinical patterns of early benefit and later hazard plateauing, and generates plausible long-term projections.

The generalised gamma (independent) model is suitable for scenario or sensitivity analysis. While more flexible, its behavior is nearly identical to log-normal in this dataset, and the added complexity offers no clear advantage.

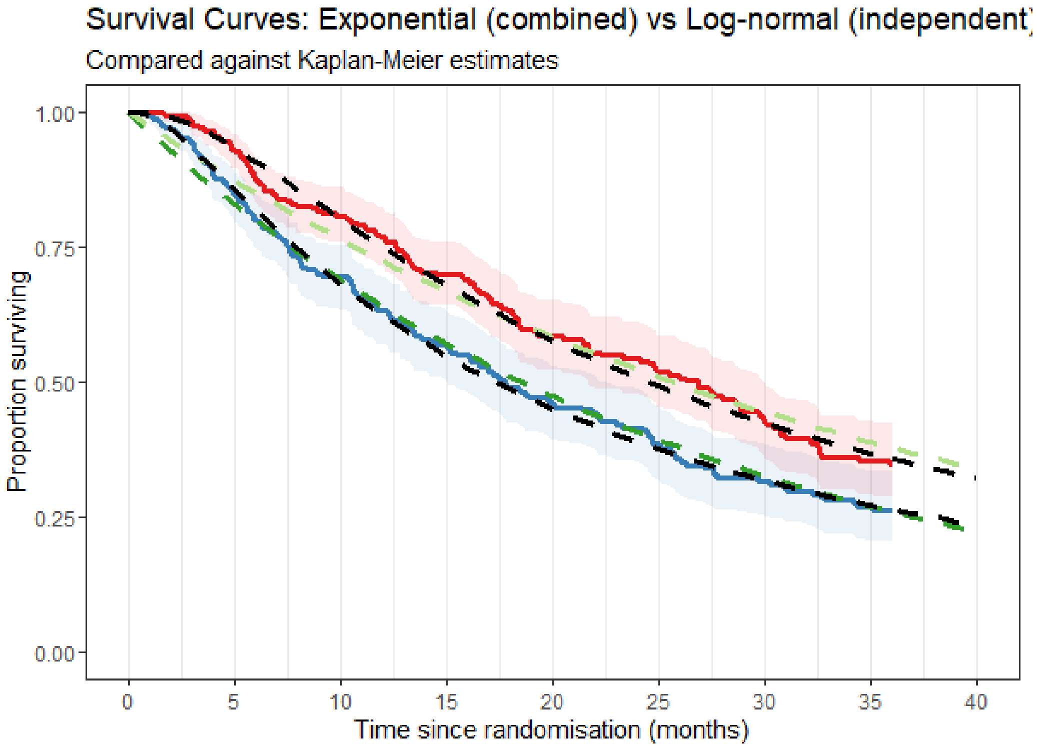
Given observed hazard flattening and long-term non-progression in some patients, future work may explore flexible parametric model (appendix 2), spline-based models (e.g., Royston-Parmar) or cure models, which offer greater structural flexibility and may better capture heterogeneity in extrapolated survival.

1.5 What is the mean expected PFS benefit for your chosen model? Compare this to the mean expected PFS benefit for the combined exponential model calculated in 1.3 and discuss. [8 marks]

Based on the recommended log-normal (independent) model, the estimated mean progression-free survival (PFS) was 46.7 months for Ultradrug and 31.7 months for control, yielding a 15.0-month benefit.
In contrast, the combined exponential model estimated 37.1 months for Ultradrug and 26.8 months for control, with a 10.3-month difference which is 4.7 months less than log-normal.

This discrepancy reflects fundamental structural differences. The exponential model's assumption of constant hazard leads to flat extrapolation, underestimating survival between 10-30 months where Kaplan-Meier curves show clear separation (Figure 17 & 18). It also fails to capture hazard plateauing after 24 months as treatment effect wanes (Figure 19).

Figure 17 Short-term Survival Curves: Exponential (Combined) vs. Log-normal (Independent)



Blue solid line: Control - *KM*: Kaplan–Meier survival estimates for the control group

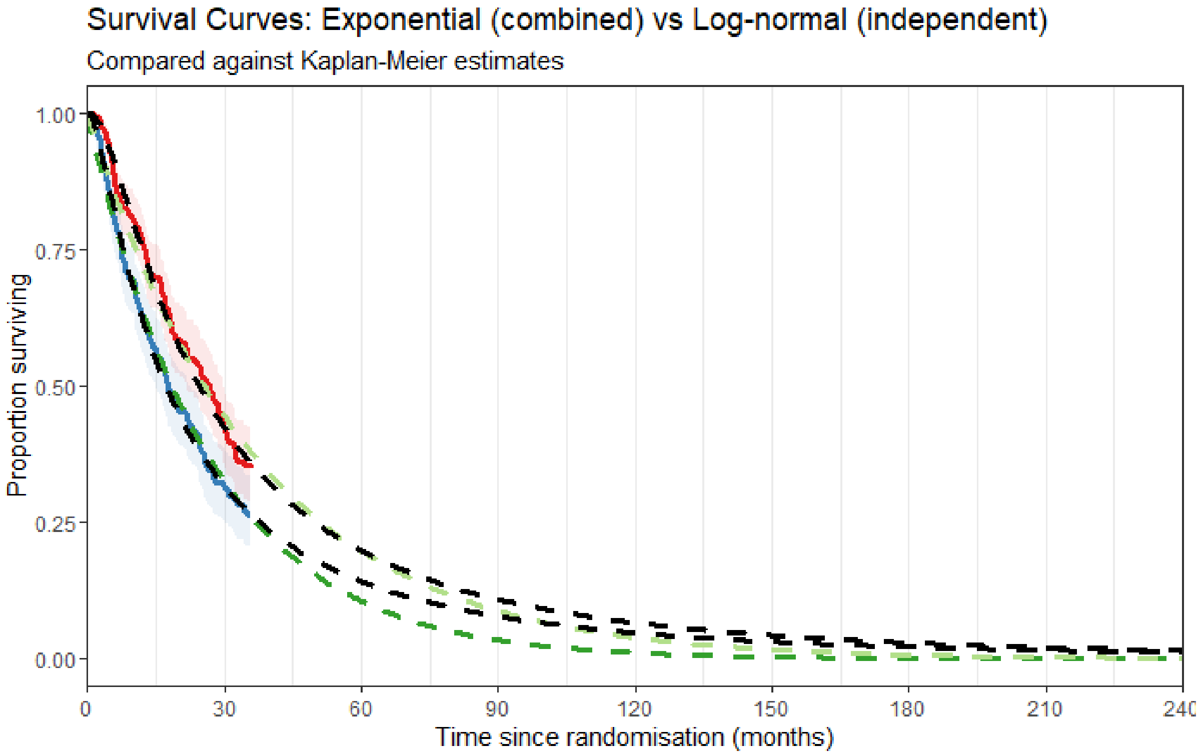
Red solid line: Experimental - *KM*: Kaplan–Meier survival estimates for the Ultradrug group

Green dashed line: *Control - Exponential*: Exponential model fit for the control group

Light green dashed line: *Experimental - Exponential*: Exponential model fit for the Ultradrug group

Black dashed line: Control *& Experimental - Log-normal*: Log-normal model fits for both groups (same color)

Figure 18 Long-term Survival Curves: Exponential (Combined) vs. Log-normal (Independent)



Blue solid line: Control - *KM*: Kaplan–Meier survival estimates for the control group

Red solid line: Experimental - *KM*: Kaplan–Meier survival estimates for the Ultradrug group

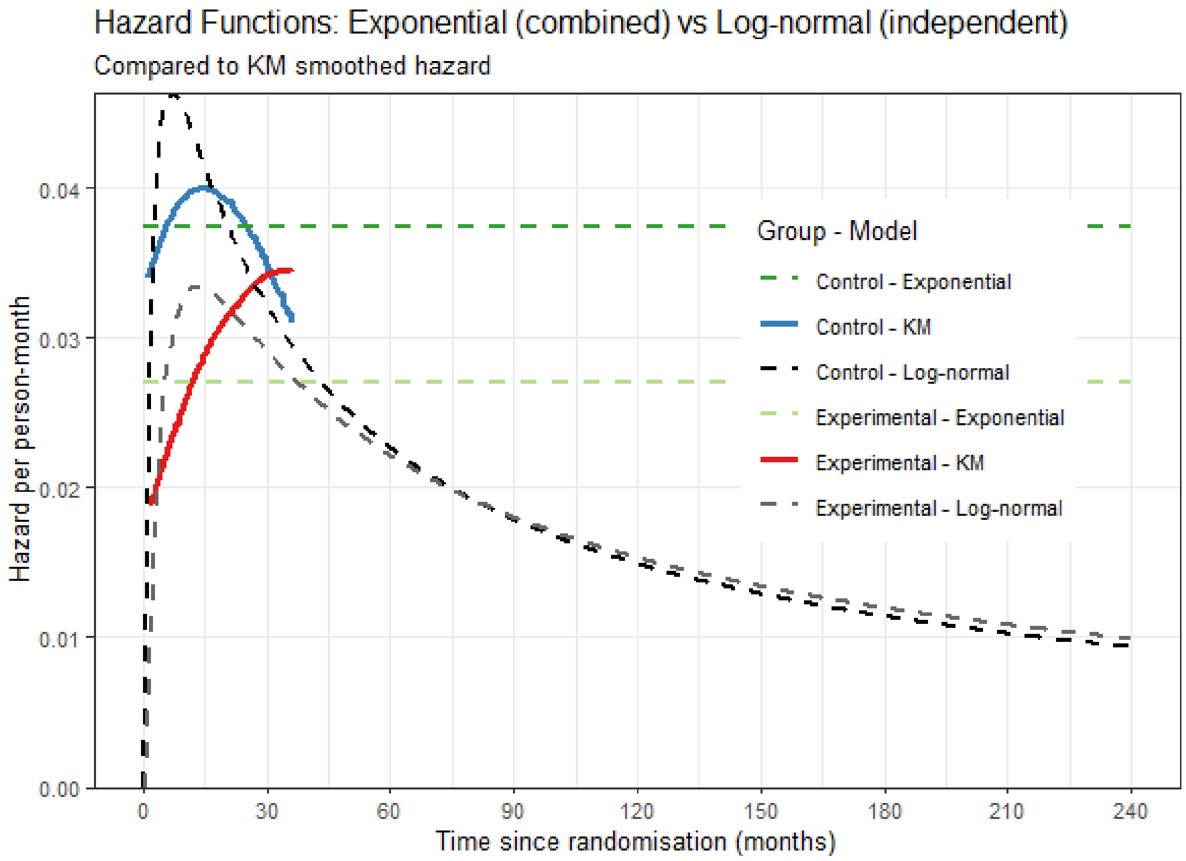
Green dashed line: *Control - Exponential*: Exponential model fit for the control group

Light green dashed line: *Experimental - Exponential*: Exponential model fit for the Ultradrug group

Black dashed line: Control *& Experimental - Log-normal*: Log-normal model fits for both groups (same color)

The log-normal model accommodates non-monotonic hazards: the hazard rises initially, reflecting early efficacy, then flattens, aligning with treatment waning and potential disease stabilization. Its extended tail suggests some patients remain progression-free long term consistent with clinical expectations and observed trends, producing a more realistic survival estimate.

Figure 19 Hazard Functions: Exponential (Combined) vs. Log-normal (Independent)



In economic terms, mean PFS directly impacts QALYs and progression-related costs. Underestimating long-term survival as with the exponential model can inflate ICERs and undervalue treatment. The log-normal model, by reflecting clinical dynamics, offers a more credible basis for base-case extrapolation.

Section 2: Treatment Switching

2.1 For the purpose of a submission to a health technology assessment agency, which of these treatments may need to be adjusted for, and why? [5 marks]

In health technology assessment (HTA), adjusting for treatment switching is necessary when patients in the control arm receive the intervention under evaluation or other non-standard therapies, thereby distorting the intended comparison. In this trial, 75 patients in the control group received Ultradrug after disease progression. Since Ultradrug is not currently available as a standard post-progression therapy, this switching artificially improves survival in the control group and biases the estimated treatment effect towards the null. Adjustment is required to avoid underestimating the benefit of Ultradrug.

In addition, 30 patients in the Ultradrug arm received Superamab post-progression, which is likewise unavailable in current clinical practice. This may lead to upward bias in the treatment group's overall survival and warrants adjustment, especially in sensitivity analysis. Conversely, Standalabine and Normastar are available in routine care and were administered in both trial arms. These therapies reflect standard practice and do not require adjustment. In summary, switching to Ultradrug in the control group and use of Superamab in the experimental group should be adjusted for, while no adjustment is needed for the standard therapies.

2.2 Given the characteristics of the trial, the information provided in Table 2 and the treatments you identified in Question 2.1, which adjustment methods would you consider using? Would you rule any methods out based on the information provided? Explain your answer. [10 marks]

Several methods are available to adjust for treatment switching, including Inverse Probability of Censoring Weights (IPCW), Two-Stage Estimation (TSE), and Rank Preserving Structural Failure Time Models (RPSFTM). Given the structure of this trial and the data available, TSE is the most appropriate method for primary analysis.

In this study, treatment switching occurred exclusively after disease progression, which is a well-defined and observed clinical event. This structure enables progression to be used as a secondary baseline, satisfying a key assumption for TSE. In addition, the trial collected relevant prognostic factors both at baseline and at the time of progression. These covariates can be included in an Accelerated Failure Time (AFT) model to estimate the time ratio between switchers and non-switchers, allowing reconstruction of counterfactual post-progression survival under a no-switching scenario. This makes TSE particularly well-suited to address the specific crossover from the control group to Ultradrug.

IPCW is also a technically viable method in this setting and may be used for sensitivity analysis. The trial collected time-updated performance status, which supports the modelling of switching probabilities over time. However, IPCW relies on strong assumptions, including no unmeasured confounding and sufficient covariate overlap (positivity) between switchers and non-switchers. Given the moderately high proportion of switching (75 of 153 progressed patients in the control group) and the risk of weight instability, IPCW may produce biased or imprecise estimates if these assumptions are violated.

RPSFTM is not recommended in this setting. It assumes a common treatment effect regardless of timing, which is unlikely to hold in this trial given the heterogeneity in post-progression therapies and the late introduction of Ultradrug in some control patients. Furthermore, RPSFTM does not accommodate indirect switching, such as Superamab use in the experimental group, and lacks a mechanism for adjusting time-dependent confounding. For these reasons, RPSFTM is ruled out.

2.3 Describe the main steps that would be required to apply the method(s) that you would consider using in this case, as identified in the Question 2.3. Take into account the characteristics of the trial and the information provided in Table 2.1, and explain any assumptions you would need to make [10 marks]

To implement the Two-Stage Estimation (TSE) method, the first step is to define disease progression as the secondary baseline. This is justified by the trial design, as all switching to Ultradrug occurred post-progression, and the timing of progression is known for all patients. The control group is then split into those who switched after progression and those who did not.

An Accelerated Failure Time (AFT) model is used to estimate a time ratio between switchers and non-switchers, based on post-progression survival. Covariates measured at progression such as performance status and tumour size are included to adjust for prognostic imbalance. The estimated ratio is then applied to adjust switchers' post-progression survival, generating counterfactual outcomes. These are added to observed pre-progression durations to reconstruct total survival under a no-switching scenario.

The reconstructed data for switchers are combined with observed data from non-switchers to form an adjusted control group. This dataset is then analysed using Cox or parametric models to estimate treatment effects without bias from switching.

TSE relies on several assumptions: (1) switching occurs shortly after progression; (2) relevant prognostic factors are measured at progression; and (3) the time ratio is valid across switchers. Violations such as delayed switching or unmeasured confounding can introduce bias.

Although IPCW may also be applied, its reliance on stronger assumptions and risk of unstable weights makes it more suitable for sensitivity analysis. Given the trial structure and available covariates, TSE is preferred for adjusting treatment switching in this case.

Section 3: Three State Model

3.1 You have been asked to develop a health economic model to assess the cost-effectiveness of Ultradrug versus BSC for hypothetical cancer A. Please explain the main steps that would be required to estimate the ICER for Ultradrug using a partitioned survival modelling approach. You can assume that parametric survival models have already been fitted to the available data on PFS and OS from the trial. [15 marks]

A partitioned survival model (PSM) with three mutually exclusive health states: progression-free, post-progression, and death can be serves as the framework for evaluating the cost-effectiveness of Ultradrug versus best supportive care (BSC). This structure reflects the clinical trajectory observed in the trial and is widely used in oncology health technology assessment.

Cohort distribution across health states over time is derived from parametric survival functions. The proportion of patients in the progression-free state is determined directly from the PFS curve; those in the post-progression state are estimated as the difference between OS and PFS; death is the residual. This indirect partitioning avoids the need to model explicit transitions, which are often unidentifiable in late-stage cancer trials.

PFS is modelled using a log-normal distribution with independent fits for each treatment arm, selected for its flexibility in capturing non-monotonic hazard shapes, early treatment benefit, and post-discontinuation waning. OS is estimated separately to allow adjustment for treatment switching using two-stage estimation, mitigating bias from non-standard post-progression therapies. Both curves are extrapolated over a lifetime horizon to reflect long-term outcomes.

QALYs are computed by multiplying state-specific utility weights by time spent in each health state, with annual discounting applied. Utility values distinguish between pre- and post-progression periods and can vary over time to reflect toxicity, recovery, and treatment duration. A decrement applies during Ultradrug exposure, capped at two years based on trial protocols and expert input.

Costs are accrued based on health state occupancy and treatment received. Progression-free costs include Ultradrug acquisition, administration, and monitoring. Post-progression costs encompass further therapies, supportive care, and healthcare use. All costs are estimated from an NHS and Personal Social Services (PSS) perspective using UK unit prices, with assumptions aligned to clinical practice and informed by expert advice.

Total discounted costs and QALYs are estimated for each arm. The incremental cost-effectiveness ratio (ICER) is calculated as the ratio of incremental costs to incremental QALYs.

Uncertainty is addressed through deterministic sensitivity analysis on structural and parameter assumptions, including the choice of parametric model (e.g., generalised gamma), time horizon, and utilities. Probabilistic sensitivity analysis propagates input uncertainty and quantifies variation in the ICER. Scenario analyses examine extrapolation beyond five years to reflect expert opinion that 10–15% of BSC patients may remain progression-free at that time.

Overall, the PSM structure integrates empirical survival data, extrapolated projections, and clinical insight to deliver transparent cost-effectiveness estimates suitable for HTA decision-making.

Appendix 1: R Code

```
#---------------------------------------------------------------------------------------------------- #
#******  1.1 Data set analysis *****************************************
#---------------------------------------------------------------------------------------------------- #

#-------------------------------------------------------------------------------------------#
#----- 1. Set up the environment and load required packages  ----- #
#-------------------------------------------------------------------------------------------#

rm(list = ls())

# Load required packages
packages  <- c("haven", "skimr", "survival", "survRM2", "survminer", "ggpubr",
             "muhaz", "ggplot2", "bshazard", "gridExtra", "flexsurv", "dplyr",
             "rstpm2", "splines", "survHE")
for (p in packages) {
  if (!require(p, character.only = TRUE)) install.packages(p, dependencies = TRUE)
  library(p, character.only = TRUE)
}


#---------------------------------------------------------------------------- #
#----- 2. Set working directory to where your assignment data is  ----- #
#---------------------------------------------------------------------------- #

setwd("C:/Users/user/Desktop/0611 Further Stat")

# Check if the directory has been correctly assigned.
getwd()

# Ensure a subfolder exist for storing figures under folder where you stored the data
# (it will automatically create the figure folder for you if you don't have it at the path where
you store the data)
if (!dir.exists("figures")) {dir.create("figures", recursive = TRUE)}

#------------------------------ #
#----- 3. Open the dataset ----- #
#------------------------------ #

# Read CSV file of data
data <- read.csv("assignment_2_data.csv")

#---------------------------------------------------------------------------- #
#-----  4. Look at the dataset ---------------------- #
#---------------------------------------------------------------------------- #
```

```
View(data)
head(data, 10)
skim(data)
summary(data)

# treatment group distribution
summary(as.factor(data$trtgrp))


# create numeric event variables：1 = progressed/died, 0 = alive
data$event_num <- ifelse(data$event == "progressed/died", 1, 0)

# create numeric cens variables：1 = censored, 0 = progressed/died
data$cens_num <- ifelse(data$cens == "censored", 1, 0)

# create factor variables：0 = Control, 1 = Ultradrug
data$trtgrp_num <- ifelse(data$trtgrp == "Control group", 0, 1)
data$trtgrp_fac <- factor(data$trtgrp_num, labels = c("Control", "Ultradrug"))

#* What are some of the key characteristics of the dataset?

#-----------------------------------------------------------------------------------------------------------------------
#
#-----  5. Obtain the Kaplan-Meier survivor function for PFS for each treatment group ---------
------------- #
#-----------------------------------------------------------------------------------------------------------------------
#
# When creating the survival object, also convert PFS_days to months for better
interpretation in graph (divide PFS_days by 30.4375)
data$PFS_months <- data$PFS_days / 30.4375 # create a new column that represent PFS
in months

surv_obj <- Surv(time = data$PFS_months, # Survival times in months
              event = data$event_num) # Event indicator

# Display the first 20 patient's survival time
head(surv_obj, 20)

# Fit Kaplan-Meier survival curves by treatment group using the surv_obj we just created
# Specify treatment group indicator as a factor (i.e., factor(trtgrp_fac)) after ~ in the survfit()
function
km_fit <- survfit(surv_obj ~ trtgrp_fac, data = data, type = "kaplan-meier")
km_fit

# Print Kaplan-Meier survival estimates with 95% CI (equivalent to STATA's 'sts list')
summary(km_fit)
print(summary(km_fit), digits = 4)
```

```
# Print KM estimates at specified times e.g., at month 0 to 5
summary(km_fit, c(0,1,2,3,4,5))

#--------------------------------------------------------------------------------------------------------------------
------------ #
#-----  6. What is the median survival time in each treatment group? What is the restricted
mean survival time? ---------------------- #
#--------------------------------------------------------------------------------------------------------------------
------------ #

# Median survival time can be read from the KM survivor function obtained in the previous
question
print(km_fit, digits = 5)

# Compute RMST & 95%CI for each treatment group using the rmst2() function of the
survRM2 package
rmst2(time = data$PFS_months, status = data$event_num,
     arm = data$trtgrp_num) # Treatment group indicator as numeric

#-----------------------------------------------------------------------------------------------#
#-----  7. Plot the Kaplan-Meier survival curves for each treatment group ---------------------- #
#---------------------------------------------------------------------------------------------- #

# Basic KM plot
ggsurvplot(km_fit, data = data)

# KM plot with nicer options and risk table
surv_plot_km <- ggsurvplot(km_fit, data = data,
                      xlab = "Time (months)",
                      ylab = "Progression-free survival probability",
                      title = "Kaplan-Meier PFS estimates",
                      legend.title = "Treatment Group",
                      legend.labs = c("Control", "Ultradrug"),
                      palette = c("#377EB8", "#E41A1C"),
                      size = 0.8,
                      censor = FALSE,
                      conf.int = TRUE,
                      risk.table = TRUE,
                      risk.table.title = "Number at risk",
                      risk.table.y.text = FALSE,
                      risk.table.height = 0.25,
                      break.time.by = 2,
                      ggtheme = theme_bw())

surv_plot_km
```

```
#------------------------------------------------------------------------------------------------- #
#******  1.2 Test propotional hazard assumption ****************************************


#---------------------------------------------------------------------------------------------- #
#-----  1. Conduct a log-rank test. --------------------------------------------------------------- #
#---------------------------------------------------------------------------------------------- #
# Log-rank test
survdiff(surv_obj ~ trtgrp_fac, data = data)

#------------------------------------------------------------------------------------------------- #
#-----  2. Plot the hazard function and the cumulative hazard function for each treatment
group ------- #
#------------------------------------------------------------------------------------------------- #
# Control group
hazard0_bshazard <- bshazard(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 0, ])
# Ultradrug group
hazard1_bshazard <- bshazard(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 1, ])

# Smoothed hazard plot
smoothed_hazard <- ggplot() +
  geom_ribbon(aes(x = hazard0_bshazard$time, ymin = hazard0_bshazard$lower.ci, ymax =
hazard0_bshazard$upper.ci, fill = "Control"), alpha = 0.2) +
  geom_ribbon(aes(x = hazard1_bshazard$time, ymin = hazard1_bshazard$lower.ci, ymax =
hazard1_bshazard$upper.ci, fill = "Ultradrug"), alpha = 0.2) +
  geom_line(aes(x = hazard0_bshazard$time, y = hazard0_bshazard$hazard, color =
"Control"), size = 1) +
  geom_line(aes(x = hazard1_bshazard$time, y = hazard1_bshazard$hazard, color =
"Ultradrug"), size = 1) +
  labs(title = "Smoothed Hazard Estimates (bshazard)", x = "Time (Months)", y = "Hazard
Rate") +
  scale_color_manual(values = c("Control" = "#377EB8", "Ultradrug" = "#E41A1C")) +
  scale_fill_manual(values = c("Control" = "#377EB8", "Ultradrug" = "#E41A1C")) +
  theme_minimal()

smoothed_hazard
ggsave("figures/Smoothed_hazard_PFS.jpg", plot = smoothed_hazard, width = 8, height = 6,
dpi = 500)

# Cumulative hazard plot
cumhaz_plot <- ggsurvplot(km_fit, data = data, fun = "cumhaz",
                        xlab = "Time (Months)", ylab = "Cumulative Hazard",
                        title = "Cumulative hazard plot",
                        legend.title = "Treatment Group",
                     legend.labs = c("Control", "Ultradrug"),
                     conf.int = TRUE, censor = FALSE,
                     palette = c("#377EB8", "#E41A1C"),
                     ggtheme = theme_minimal())

cumhaz_plot
ggexport(cumhaz_plot, filename = "figures/Cumulative_hazard_PFS.jpg", width = 800,
height = 600, dpi = 1000)

#--------------------------------------------------------------------------------------------------- #
#-----  3. Fit a Cox proportional hazards model to estimate the treatment effect.  -- #
#--------------------------------------------------------------------------------------------------- #

fit_cox <- coxph(surv_obj ~ trtgrp_fac, data = data)
summary(fit_cox)

#--------------------------------------------------------------------------------------------------- #
#-----  4. Complementary log-log plot ------------------------------------------------------------ #
#--------------------------------------------------------------------------------------------------- #

cloglog_plot <- ggsurvplot(km_fit, data = data, fun = "cloglog",
                      xlab = "Time (Months)",
                      title = "Complementary log-log plot",
                      legend.title = "Treatment Group",
                      legend.labs = c("Control", "Ultradrug"),
                      censor = FALSE,
                      ggtheme = theme_bw(),
                      palette = c("#377EB8", "#E41A1C"))

cloglog_plot
ggexport(cloglog_plot, filename = "figures/Complementary_log_plot_PFS.jpg", width = 800,
height = 600, dpi = 1000)

#--------------------------------------------------------------------------------------------------- #
#-----  5. Test the Schoenfeld residuals to assess proportional hazards ------------ #
#--------------------------------------------------------------------------------------------------- #
cox_zph <- cox.zph(fit_cox, transform = "identity")
cox_zph

# Plot Schoenfeld residuals
Schoenfeld_residuals <- ggcoxzph(cox_zph)
Schoenfeld_residuals

ggsave("figures/Schoenfeld_residuals_PFS.jpg", arrangeGrob(grobs =
Schoenfeld_residuals), width = 10, height = 6, dpi = 300)

# Optional: Q-Q plot to assess constant time ratio assumption (relevant for AFT models)
```

```
qqplot(data$PFS_days[data$trtgrp_num == 0],
       data$PFS_days[data$trtgrp_num == 1],
       main = "Quantile-Quantile (Q-Q) Plot",
       xlab = "Control Group PFS Time Quantiles",
       ylab = "Ultradrug Group PFS Time Quantiles")
abline(0, 1, col = "red", lwd = 2)  # Reference line


#----------------------------------------------------------------------------------------------------- #
#-----1.3 Fit exponential parametric models (combined) for PFS ---------------------- #
#----------------------------------------------------------------------------------------------------- #

#****** Exponential - treatment group as a covariate ("combined")
******************************************

fit_Exponential<- flexsurvreg(surv_obj ~ trtgrp_num, data = data, dist = "exp")
fit_Exponential

# AIC and BIC
AIC(fit_Exponential)
BIC(fit_Exponential)

# Mean survival time
mean_exp_con <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "mean", newdata = data.frame(trtgrp_num = 0)), `[[`,
"est")))
mean_exp_exp <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "mean", newdata = data.frame(trtgrp_num = 1)), `[[`,
"est")))

# Predicted survival functions
t_predict <- seq(0, 240, by = 0.5)

surv_exp_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num
= 0)), `[[`, "est")))
surv_exp_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num
= 1)), `[[`, "est")))

# Predicted hazard functions
haz_exp_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num =
0)), `[[`, "est")))
haz_exp_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num =
1)), `[[`, "est")))
```

```
#****** Plot Survival curves: Exponential model vs KM curves
******************************************
km_fit0 <- survfit(Surv(PFS_months, event_num) ~ 1, data = data[data$trtgrp_num == 0, ])
km_fit1 <- survfit(Surv(PFS_months, event_num) ~ 1, data = data[data$trtgrp_num == 1, ])

# Prepare KM curves for plotting
km_surv_con <- data.frame(time    = c(0, km_fit0$time),
                          surv    = c(1, km_fit0$surv),
                          upper   = c(1, km_fit0$upper),
                          lower   = c(1, km_fit0$lower))

km_surv_exp <- data.frame(time    = c(0, km_fit1$time),
                          surv    = c(1, km_fit1$surv),
                          upper   = c(1, km_fit1$upper),
                          lower   = c(1, km_fit1$lower))

# Plot survival curves
Survival_Exponential_36m <- ggplot() +

  # KM 95% CI ribbons
  geom_ribbon(aes(x = km_surv_con$time, ymin = km_surv_con$lower, ymax =
km_surv_con$upper, fill = "Control - 95% CI"), alpha = 0.1) +
  geom_ribbon(aes(x = km_surv_exp$time, ymin = km_surv_exp$lower, ymax =
km_surv_exp$upper, fill = "Treatment - 95% CI"), alpha = 0.1) +

  # Exponential survival curves
  geom_line(aes(x = t_predict, y = surv_exp_comb_con, color = "Control - Exponential"),
linewidth = 1) +
  geom_line(aes(x = t_predict, y = surv_exp_comb_exp, color = "Treatment - Exponential"),
linetype = "dashed", linewidth = 1) +

  # KM step curves
  geom_step(aes(x = km_surv_con$time, y = km_surv_con$surv, color = "Control - KM"),
linewidth = 1) +
  geom_step(aes(x = km_surv_exp$time, y = km_surv_exp$surv, color = "Treatment - KM"),
linewidth = 1) +

  # Colors and fills
  scale_color_manual(values = c("Control - KM" = "#377EB8",
                     "Treatment - KM" = "#E41A1C",
                     "Control - Exponential" = "blue",
                     "Treatment - Exponential" = "blue")) +
  scale_fill_manual(values = c("Control - 95% CI" = "#377EB8",
                    "Treatment - 95% CI" = "#E41A1C"), guide = "none") +

  # Labels and theme
  labs(title = "Exponential Model vs Kaplan-Meier Survival Estimates",
       x = "Time since randomisation (months)",
       y = "Proportion surviving",
       color = "Group",
       fill = "Group") +

  scale_x_continuous(limits = c(0, 36), breaks=seq(0, 36, by =3), expand = c(0, 0.05)) +
  scale_y_continuous(limits = c(0, 1), breaks = seq(0, 1, by = 0.1), expand = c(0.02, 0)) +

  theme_bw() +
  theme(legend.position = c(0.5, 0.8),
        text = element_text(size = 14),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank()) +
  guides(color = guide_legend(ncol = 2))

# Display plot
Survival_Exponential_36m

# Extrapolate to 240 months
Survival_Exponential240m <- Survival_Exponential_36m +
  scale_x_continuous(limits = c(0, 240), breaks = seq(0, 240, by = 30), expand = c(0, 0.05))

#****** Plot Hazard curves: Exponential model vs smoothed hazard
*******************************************

Hazard_Exponential_36m <- ggplot() +

  # Smoothed hazard CI ribbons
  geom_ribbon(aes(x = hazard0_bshazard$time, ymin = hazard0_bshazard$lower.ci, ymax =
hazard0_bshazard$upper.ci, fill = "Control - 95% CI"), alpha = 0.1) +
  geom_ribbon(aes(x = hazard1_bshazard$time, ymin = hazard1_bshazard$lower.ci, ymax =
hazard1_bshazard$upper.ci, fill = "Treatment - 95% CI"), alpha = 0.1) +

  # Exponential hazard curves
  geom_line(aes(x = t_predict[-1], y = haz_exp_comb_con[-1], color = "Control -
Exponential"), linewidth = 1) +
  geom_line(aes(x = t_predict[-1], y = haz_exp_comb_exp[-1], color = "Treatment -
Exponential"), linetype = "dashed", linewidth = 1) +

  # Smoothed hazard step curves
  geom_step(aes(x = hazard0_bshazard$time, y = hazard0_bshazard$hazard, color =
"Control - Smoothed KM"), linewidth = 1) +
  geom_step(aes(x = hazard1_bshazard$time, y = hazard1_bshazard$hazard, color =
"Treatment - Smoothed KM"), linewidth = 1) +

  # Colors and fills
  scale_color_manual(values = c("Control - Smoothed KM" = "#377EB8",
                              "Treatment - Smoothed KM" = "#E41A1C",
                              "Control - Exponential" = "blue",
                              "Treatment - Exponential" = "blue")) +
  scale_fill_manual(values = c("Control - 95% CI" = "#377EB8",
                             "Treatment - 95% CI" = "#E41A1C"), guide = "none") +

  # Labels and theme
  labs(title = "Exponential Model vs Smoothed Hazard Estimates",
       x = "Time since randomisation (months)",
       y = "Hazard per person-month",
       color = "Group",
       fill = "Group") +

  scale_x_continuous(limits = c(0, 36), breaks=seq(0, 36, by = 3), expand = c(0, 0.05)) +
  scale_y_continuous(limits = c(0, 0.08), breaks = seq(0, 0.08, by = 0.02), expand = c(0.02,
0)) +

  theme_bw() +
  theme(legend.position = c(0.5, 0.8),
        text = element_text(size = 14),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank()) +
  guides(color = guide_legend(ncol = 2))

# Display plot
Hazard_Exponential_36m


#------------------------------------------------------------------------------------------------ #
#-----1.4 Fit other parametric models (combined and independent) for PFS -------- #
#------------------------------------------------------------------------------------------------ #



#------------------------------------------------------------------------------------------------ #
#-----STEP1 : Assess parametric model assumptions ---------------------------- #
#------------------------------------------------------------------------------------------------ #
km_fit0 <- survfit(Surv(PFS_months, event_num) ~ 1, data = data[data$trtgrp_num == 0, ])
km_fit1 <- survfit(Surv(PFS_months, event_num) ~ 1, data = data[data$trtgrp_num == 1, ])

hazf0 <- km_fit0$n.event / km_fit0$n.risk
hazf1 <- km_fit1$n.event / km_fit1$n.risk

logh0 <- log(hazf0)
logh1 <- log(hazf1)

logt0 <- log(summary(km_fit0)$time)
```

```
logt1 <- log(summary(km_fit1)$time)

survf0 <- summary(km_fit0)$surv
survf1 <- summary(km_fit1)$surv

logs0 <- log(survf0)
logs1 <- log(survf1)

minuslogs0 <- -log(survf0)
minuslogs1 <- -log(survf1)

minus2logs0 <- -log(minuslogs0)
minus2logs1 <- -log(minuslogs1)

logoddss0 <- log(survf0 / (1 - survf0))
logoddss1 <- log(survf1 / (1 - survf1))

invnormals0 <- qnorm(1 - survf0)
invnormals1 <- qnorm(1 - survf1)

# Weibull / exponential
ggplot() +
  geom_point(aes(x = logt0, y = -minus2logs0, color = "Control")) +
  geom_point(aes(x = logt1, y = -minus2logs1, color = "Ultradrug")) +
  labs(x = "log(t)", y = "log(-log(S(t)))",
       title = "AFT Model Assessment: Weibull/Exponential (PFS)", color = "Group") +
  theme_bw() +
  scale_color_manual(values = c("Control" = "#377EB8", "Ultradrug" = "#E41A1C"))
# Log-logistic
log_logistic_logplot <- ggplot() +
  geom_point(aes(x = logt0, y = logoddss0, color = "Control")) +
  geom_point(aes(x = logt1, y = logoddss1, color = "Ultradrug")) +
  labs(x = "log(t)", y = "log(S(t) / (1 - S(t)))",
       title = "AFT Model Assessment: Log-Logistic", color = "Group") +
  theme_bw() +
  scale_color_manual(values = c("Control" = "#377EB8", "Ultradrug" = "#E41A1C"))

log_logistic_logplot
ggsave("figures/log_logistic_log_plot_PFS.jpg", plot = log_logistic_logplot, width = 8, height
= 6, dpi = 300)

# Log-normal
log_normal_logplot <- ggplot() +
  geom_point(aes(x = logt0, y = invnormals0, color = "Control")) +
  geom_point(aes(x = logt1, y = invnormals1, color = "Ultradrug")) +
  labs(x = "log(t)", y = "Inv.normal(1-S(t))",
       title = "AFT Model Assessment: Log-Normal", color = "Group") +
  theme_bw() +
  scale_color_manual(values = c("Control" = "#377EB8", "Ultradrug" = "#E41A1C"))

log_normal_logplot
ggsave("figures/log_normal_log_plot_PFS.jpg", plot = log_normal_logplot, width = 8, height
= 6, dpi = 300)

# Gompertz (raw log(hazard))
ggplot() +
  geom_point(aes(x = km_fit0$time[!is.infinite(logh0)], y = logh0[!is.infinite(logh0)], color =
"Control")) +
  geom_point(aes(x = km_fit1$time[!is.infinite(logh1)], y = logh1[!is.infinite(logh1)], color =
"Ultradrug")) +
  labs(x = "Time (months)", y = "log(hazard function)",
       title = "AFT Model Assessment: Gompertz", color = "Group") +
  theme_bw() +
  scale_color_manual(values = c("Control" = "#377EB8", "Ultradrug" = "#E41A1C"))


# Gompertz (raw log(hazard))
# Step 1: Create adjusted PFS_days2 to avoid ties

data2 <- data
data2 <- data2[order(data2$event_num, data2$trtgrp_num, data2$PFS_days), ]

# Adjust PFS_days if ties occur (progressed/died = 1)
data2$PFS_days2 <- data2$PFS_days
data2$PFS_days2 <- ifelse(data2$PFS_days == lag(data2$PFS_days) & data2$event_num
== 1, data2$PFS_days + 0.3, data2$PFS_days2)
data2$PFS_days2 <- ifelse(data2$PFS_days2 == lag(data2$PFS_days2) &
data2$event_num == 1, data2$PFS_days2 + 0.3, data2$PFS_days2)
data2$PFS_months2 <- data2$PFS_days2 / 30.4375

# Step 2: Refit KM curves for each treatment group with adjusted PFS_months2
km_fit_adj0 <- survfit(Surv(PFS_months2, event_num) ~ 1, data = data2[data2$trtgrp_num
== 0, ], type = "kaplan-meier")
km_fit_adj1 <- survfit(Surv(PFS_months2, event_num) ~ 1, data = data2[data2$trtgrp_num
== 1, ], type = "kaplan-meier")

# Step 3: Calculate hazard and log(hazard)
hazf_adj0 <- km_fit_adj0$n.event / km_fit_adj0$n.risk
hazf_adj1 <- km_fit_adj1$n.event / km_fit_adj1$n.risk

logh_adj0 <- log(hazf_adj0)
logh_adj1 <- log(hazf_adj1)

# Step 4: Plot the adjusted log(hazard)
```

```
Gompertz_logplot_PFS <- ggplot() +
  geom_point(aes(x = km_fit_adj0$time[!is.infinite(logh_adj0)], y =
logh_adj0[!is.infinite(logh_adj0)], color = "Control")) +
  geom_point(aes(x = km_fit_adj1$time[!is.infinite(logh_adj1)], y =
logh_adj1[!is.infinite(logh_adj1)], color = "Ultradrug")) +
  labs(x = "Time (months)", y = "log(hazard function)",
       title = "AFT Model Assessment: Gompertz (Adjusted for Ties) — PFS", color = "Group")
+
  theme_bw() +
  scale_color_manual(values = c("Control" = "#377EB8", "Ultradrug" = "#E41A1C"))

# Display plot
Gompertz_logplot_PFS


#---------------------------------------------------------------------------------------------------- #
#-----STEP2 : Fit different parametric models------------------------------ #
#---------------------------------------------------------------------------------------------------- #

# - Exponential: dist = "exp"
# - Weibull: dist = "weibullPH"
# - Log-normal: dist = "lnorm"
# - Log-logistic: dist = "llogis"
# - Gompertz: dist = "gompertz"
# - Gamma: dist = "gamma"
# - Generalized Gamma: dist = "gengamma"


#---------------------------------------------------------------------------------------------------- #
#******  Exponential *****************************************
#---------------------------------------------------------------------------------------------------- #
#******  Exponential - treatment group as covariate ("combined")
*****************************************
fit_Exponential <- flexsurvreg(surv_obj ~ trtgrp_num, data = data, dist = "exp")
fit_Exponential

# AIC and BIC
AIC(fit_Exponential)
BIC(fit_Exponential)

# Mean survival time
mean_exp_con <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "mean", newdata = data.frame(trtgrp_num = 0)), `[[`,
"est")))

mean_exp_exp <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "mean", newdata = data.frame(trtgrp_num = 1)), `[[`,
"est")))

# Predicted survival functions
t_predict <- seq(0, 240, by = 0.5)

surv_exp_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num
= 0)), `[[`, "est")))

surv_exp_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num
= 1)), `[[`, "est")))

# Predicted hazard functions
haz_exp_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num =
0)), `[[`, "est")))

haz_exp_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Exponential, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num =
1)), `[[`, "est")))

#******   Exponential - independent model for experimental group
*******************************************
fit_Exp_ind_con <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 0, ], dist = "exp")
fit_Exp_ind_con

# AIC and BIC
AIC(fit_Exp_ind_con)
BIC(fit_Exp_ind_con)

# Mean survival time
mean_exp_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Exp_ind_con, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_exp_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Exp_ind_con, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_exp_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Exp_ind_con, type = "hazard", t = t_predict), `[[`, "est")))

#******   Exponential - independent model for experimental group
*******************************************
```

```
fit_Exp_ind_exp <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 1, ], dist = "exp")
fit_Exp_ind_exp

# AIC and BIC
AIC(fit_Exp_ind_exp)
BIC(fit_Exp_ind_exp)

# Mean survival time
mean_exp_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Exp_ind_exp, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_exp_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Exp_ind_exp, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_exp_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Exp_ind_exp, type = "hazard", t = t_predict), `[[`, "est")))

#--------------------------------------------------------------------------------------------------- #
#******  Weibull ******************************************
#--------------------------------------------------------------------------------------------------- #

#******  Weibull - treatment group as covariate ("combined")
******************************************
fit_Weibull <- flexsurvreg(surv_obj ~ trtgrp_num, data = data, dist = "weibullPH")
fit_Weibull

# AIC and BIC
AIC(fit_Weibull)
BIC(fit_Weibull)

# Mean survival time
mean_weibull_con <- as.numeric(unlist(lapply(
  summary(fit_Weibull, type = "mean", newdata = data.frame(trtgrp_num = 0)), `[[`, "est")))
mean_weibull_exp <- as.numeric(unlist(lapply(
  summary(fit_Weibull, type = "mean", newdata = data.frame(trtgrp_num = 1)), `[[`, "est")))

# Predicted survival functions
t_predict <- seq(0, 240, by = 0.5)

surv_weibull_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Weibull, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num =
0)), `[[`, "est")))
surv_weibull_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Weibull, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num =
1)), `[[`, "est")))

# Predicted hazard functions
haz_weibull_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Weibull, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num = 0)),
`[[`, "est")))
haz_weibull_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Weibull, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num = 1)),
`[[`, "est")))

#******   Weibull - independent model for control group ******************************************
fit_Weibull_ind_con <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 0, ], dist = "weibullPH")
fit_Weibull_ind_con

# AIC and BIC
AIC(fit_Weibull_ind_con)
BIC(fit_Weibull_ind_con)

# Mean survival time
mean_weibull_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Weibull_ind_con, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_weibull_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Weibull_ind_con, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_weibull_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Weibull_ind_con, type = "hazard", t = t_predict), `[[`, "est")))

#******   Weibull - independent model for experimental group
******************************************
fit_Weibull_ind_exp <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 1, ], dist = "weibullPH")
fit_Weibull_ind_exp

# AIC and BIC
AIC(fit_Weibull_ind_exp)
BIC(fit_Weibull_ind_exp)

# Mean survival time
mean_weibull_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Weibull_ind_exp, type = "mean"), `[[`, "est")))

# Predicted survival functions
```

```
surv_weibull_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Weibull_ind_exp, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_weibull_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Weibull_ind_exp, type = "hazard", t = t_predict), `[[`, "est")))



#---------------------------------------------------------------------------------------------------- #
#****** Log-normal ********************************************
#---------------------------------------------------------------------------------------------------- #

#****** Log-normal - treatment group as covariate ("combined")
********************************************
fit_Lnorm <- flexsurvreg(surv_obj ~ trtgrp_num, data = data, dist = "lnorm")
fit_Lnorm

# AIC and BIC
AIC(fit_Lnorm)
BIC(fit_Lnorm)

# Mean survival time
mean_lnorm_con <- as.numeric(unlist(lapply(
  summary(fit_Lnorm, type = "mean", newdata = data.frame(trtgrp_num = 0)), `[[`, "est")))
mean_lnorm_exp <- as.numeric(unlist(lapply(
  summary(fit_Lnorm, type = "mean", newdata = data.frame(trtgrp_num = 1)), `[[`, "est")))

# Predicted survival functions
t_predict <- seq(0, 240, by = 0.5)

surv_lnorm_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Lnorm, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num = 0)),
`[[`, "est")))
surv_lnorm_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Lnorm, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num = 1)),
`[[`, "est")))

# Predicted hazard functions
haz_lnorm_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Lnorm, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num = 0)),
`[[`, "est")))
haz_lnorm_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Lnorm, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num = 1)),
`[[`, "est")))
```

```
#******  Log-normal - independent model for control group
*****************************************
fit_Lnorm_ind_con <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 0, ], dist = "lnorm")
fit_Lnorm_ind_con

# AIC and BIC
AIC(fit_Lnorm_ind_con)
BIC(fit_Lnorm_ind_con)

# Mean survival time
mean_lnorm_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Lnorm_ind_con, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_lnorm_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Lnorm_ind_con, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_lnorm_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Lnorm_ind_con, type = "hazard", t = t_predict), `[[`, "est")))

#******  Log-normal - independent model for experimental group
*****************************************
fit_Lnorm_ind_exp <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 1, ], dist = "lnorm")
fit_Lnorm_ind_exp

# AIC and BIC
AIC(fit_Lnorm_ind_exp)
BIC(fit_Lnorm_ind_exp)

# Mean survival time
mean_lnorm_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Lnorm_ind_exp, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_lnorm_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Lnorm_ind_exp, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_lnorm_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Lnorm_ind_exp, type = "hazard", t = t_predict), `[[`, "est")))

#-------------------------------------------------------------------------------------------------- #
#******  Log-logistic *******************************************
#-------------------------------------------------------------------------------------------------- #
```

```
#******  Log-logistic - treatment group as covariate ("combined")
******************************************
fit_logl <- flexsurvreg(surv_obj ~ trtgrp_num, data = data, dist = "llogis")
fit_logl

# AIC and BIC
AIC(fit_logl)
BIC(fit_logl)

# Mean survival time
mean_logl_con <- as.numeric(unlist(lapply(
  summary(fit_logl, type = "mean", newdata = data.frame(trtgrp_num = 0)), `[[`, "est")))
mean_logl_exp <- as.numeric(unlist(lapply(
  summary(fit_logl, type = "mean", newdata = data.frame(trtgrp_num = 1)), `[[`, "est")))

# Predicted survival functions
t_predict <- seq(0, 240, by = 0.5)

surv_logl_comb_con <- as.numeric(unlist(lapply(
  summary(fit_logl, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num = 0)),
`[[`, "est")))
surv_logl_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_logl, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num = 1)),
`[[`, "est")))

# Predicted hazard functions
haz_logl_comb_con <- as.numeric(unlist(lapply(
  summary(fit_logl, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num = 0)), `[[`,
"est")))
haz_logl_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_logl, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num = 1)), `[[`,
"est")))

#******   Log-logistic - independent model for control group
******************************************
fit_logl_ind_con <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 0, ], dist = "llogis")
fit_logl_ind_con

# AIC and BIC
AIC(fit_logl_ind_con)
BIC(fit_logl_ind_con)

# Mean survival time
mean_logl_ind_con <- as.numeric(unlist(lapply(
  summary(fit_logl_ind_con, type = "mean"), `[[`, "est")))
```

```
# Predicted survival functions
surv_logl_ind_con <- as.numeric(unlist(lapply(
  summary(fit_logl_ind_con, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_logl_ind_con <- as.numeric(unlist(lapply(
  summary(fit_logl_ind_con, type = "hazard", t = t_predict), `[[`, "est")))

#******   Log-logistic - independent model for experimental group
*****************************************
fit_logl_ind_exp <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 1, ], dist = "llogis")
fit_logl_ind_exp

# AIC and BIC
AIC(fit_logl_ind_exp)
BIC(fit_logl_ind_exp)

# Mean survival time
mean_logl_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_logl_ind_exp, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_logl_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_logl_ind_exp, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_logl_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_logl_ind_exp, type = "hazard", t = t_predict), `[[`, "est")))

#---------------------------------------------------------------------------------------------------- #
#******  Gompertz ********************************************
#---------------------------------------------------------------------------------------------------- #

#******  Gompertz - treatment group as covariate ("combined")
*****************************************
fit_Gompertz <- flexsurvreg(surv_obj ~ trtgrp_num, data = data, dist = "gompertz")
fit_Gompertz

# AIC and BIC
AIC(fit_Gompertz)
BIC(fit_Gompertz)

# Mean survival time
mean_gompertz_con <- as.numeric(unlist(lapply(
  summary(fit_Gompertz, type = "mean", newdata = data.frame(trtgrp_num = 0)), `[[`, "est")))
```

```
mean_gompertz_exp <- as.numeric(unlist(lapply(
  summary(fit_Gompertz, type = "mean", newdata = data.frame(trtgrp_num = 1)), `[[`, "est")))

# Predicted survival functions
t_predict <- seq(0, 240, by = 0.5)

surv_gompertz_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Gompertz, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num =
0)), `[[`, "est")))
surv_gompertz_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Gompertz, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num =
1)), `[[`, "est")))

# Predicted hazard functions
haz_gompertz_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Gompertz, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num =
0)), `[[`, "est")))
haz_gompertz_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Gompertz, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num =
1)), `[[`, "est")))


#******   Gompertz - independent model for control group
********************************************
fit_Gompertz_ind_con <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 0, ], dist = "gompertz")
fit_Gompertz_ind_con

# AIC and BIC
AIC(fit_Gompertz_ind_con)
BIC(fit_Gompertz_ind_con)

# Mean survival time
mean_gompertz_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Gompertz_ind_con, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_gompertz_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Gompertz_ind_con, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_gompertz_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Gompertz_ind_con, type = "hazard", t = t_predict), `[[`, "est")))

#******   Gompertz - independent model for experimental group
********************************************
```

```
fit_Gompertz_ind_exp <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 1, ], dist = "gompertz")
fit_Gompertz_ind_exp

# AIC and BIC
AIC(fit_Gompertz_ind_exp)
BIC(fit_Gompertz_ind_exp)

# Mean survival time
mean_gompertz_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Gompertz_ind_exp, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_gompertz_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Gompertz_ind_exp, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_gompertz_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Gompertz_ind_exp, type = "hazard", t = t_predict), `[[`, "est")))

#---------------------------------------------------------------------------------------------------- #
#****** Gamma ******************************************
#---------------------------------------------------------------------------------------------------- #

#****** Gamma - treatment group as covariate ("combined")
******************************************
fit_Gamma<- flexsurvreg(surv_obj ~ trtgrp_num, data = data, dist = "gamma")
fit_Gamma

# AIC and BIC
AIC(fit_Gamma)
BIC(fit_Gamma)

# Mean survival time
mean_gamma_con <- as.numeric(unlist(lapply(
  summary(fit_Gamma, type = "mean", newdata = data.frame(trtgrp_num = 0)), `[[`, "est")))
mean_gamma_exp <- as.numeric(unlist(lapply(
  summary(fit_Gamma, type = "mean", newdata = data.frame(trtgrp_num = 1)), `[[`, "est")))

# Predicted survival functions
t_predict <- seq(0, 240, by = 0.5)

surv_gamma_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Gamma, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num =
0)), `[[`, "est")))
surv_gamma_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Gamma, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num = 
1)), `[[`, "est")))

# Predicted hazard functions
haz_gamma_comb_con <- as.numeric(unlist(lapply(
  summary(fit_Gamma, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num = 
0)), `[[`, "est")))
haz_gamma_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_Gamma, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num = 
1)), `[[`, "est")))

#****** Gamma - independent model for control group ******************************************

fit_Gamma_ind_con <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data = 
data[data$trtgrp_num == 0, ], dist = "gamma")
fit_Gamma_ind_con

# AIC and BIC
AIC(fit_Gamma_ind_con)
BIC(fit_Gamma_ind_con)

# Mean survival time
mean_gamma_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Gamma_ind_con, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_gamma_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Gamma_ind_con, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_gamma_ind_con <- as.numeric(unlist(lapply(
  summary(fit_Gamma_ind_con, type = "hazard", t = t_predict), `[[`, "est")))

#****** Gamma - independent model for experimental group 
******************************************

fit_Gamma_ind_exp <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data = 
data[data$trtgrp_num == 1, ], dist = "gamma")
fit_Gamma_ind_exp

# AIC and BIC
AIC(fit_Gamma_ind_exp)
BIC(fit_Gamma_ind_exp)

# Mean survival time
mean_gamma_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Gamma_ind_exp, type = "mean"), `[[`, "est")))
```

```
# Predicted survival functions
surv_gamma_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Gamma_ind_exp, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_gamma_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_Gamma_ind_exp, type = "hazard", t = t_predict), `[[`, "est")))


#------------------------------------------------------------------------------------------------------- #
#******  Generalized Gamma ******************************************
#------------------------------------------------------------------------------------------------------- #

#******  Generalized Gamma - treatment group as covariate ("combined")
******************************************
fit_GG <- flexsurvreg(surv_obj ~ trtgrp_num, data = data, dist = "gengamma")
fit_GG

# AIC and BIC
AIC(fit_GG)
BIC(fit_GG)

# Mean survival time
# Note: Sometimes gengamma mean may not compute due to complex hazard shape (same
as Gompertz).
# If it fails, it is acceptable to just proceed with survival and hazard plots.

mean_GG_con <- as.numeric(unlist(lapply(
  summary(fit_GG, type = "mean", newdata = data.frame(trtgrp_num = 0)), `[[`, "est")))

mean_GG_exp <- as.numeric(unlist(lapply(
  summary(fit_GG, type = "mean", newdata = data.frame(trtgrp_num = 1)), `[[`, "est")))

# Predicted survival functions
t_predict <- seq(0, 240, by = 0.5)

surv_GG_comb_con <- as.numeric(unlist(lapply(
  summary(fit_GG, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num = 0)),
`[[`, "est")))

surv_GG_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_GG, type = "survival", t = t_predict, newdata = data.frame(trtgrp_num = 1)),
`[[`, "est")))

# Predicted hazard functions
haz_GG_comb_con <- as.numeric(unlist(lapply(
  summary(fit_GG, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num = 0)), `[[`,
"est")))

haz_GG_comb_exp <- as.numeric(unlist(lapply(
  summary(fit_GG, type = "hazard", t = t_predict, newdata = data.frame(trtgrp_num = 1)), `[[`,
"est")))

#******   Generalized Gamma - independent model for control group
********************************************
fit_GG_ind_con <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 0, ], dist = "gengamma")
fit_GG_ind_con

# AIC and BIC
AIC(fit_GG_ind_con)
BIC(fit_GG_ind_con)

# Mean survival time
mean_GG_ind_con <- as.numeric(unlist(lapply(
  summary(fit_GG_ind_con, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_GG_ind_con <- as.numeric(unlist(lapply(
  summary(fit_GG_ind_con, type = "survival", t = t_predict), `[[`, "est")))

# Predicted hazard functions
haz_GG_ind_con <- as.numeric(unlist(lapply(
  summary(fit_GG_ind_con, type = "hazard", t = t_predict), `[[`, "est")))

#******   Generalized Gamma - independent model for experimental group
********************************************
fit_GG_ind_exp <- flexsurvreg(Surv(PFS_months, event_num) ~ 1, data =
data[data$trtgrp_num == 1, ], dist = "gengamma")
fit_GG_ind_exp

# AIC and BIC
AIC(fit_GG_ind_exp)
BIC(fit_GG_ind_exp)

# Mean survival time
mean_GG_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_GG_ind_exp, type = "mean"), `[[`, "est")))

# Predicted survival functions
surv_GG_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_GG_ind_exp, type = "survival", t = t_predict), `[[`, "est")))
```

```
# Predicted hazard functions
haz_GG_ind_exp <- as.numeric(unlist(lapply(
  summary(fit_GG_ind_exp, type = "hazard", t = t_predict), `[[`, "est")))

#--------------------------------------------------------------------------------------------- #
#-----STEP3 : Plotting--------------------------- #
#--------------------------------------------------------------------------------------------- #


#--------------------------------------------------------------------------------------------- #
#******  Construct survival plots *******************************************
#--------------------------------------------------------------------------------------------- #
Survival_extrapolations_all_models <- ggplot() +

  # 95% CI for Kaplan-Meier Curves
  geom_ribbon(aes(x = km_surv_con$time, ymin = km_surv_con$lower, ymax =
km_surv_con$upper, fill = "Control - 95% CI"), alpha = 0.1) +
  geom_ribbon(aes(x = km_surv_exp$time, ymin = km_surv_exp$lower, ymax =
km_surv_exp$upper, fill = "Experimental - 95% CI"), alpha = 0.1) +

  # Exponential
  geom_line(aes(x = t_predict, y = surv_exp_comb_con, color = "Control - Exponential"),
linewidth = 1) +
  geom_line(aes(x = t_predict, y = surv_exp_comb_exp, color = "Experimental -
Exponential"), linetype = "dashed", linewidth = 1) +

  # Weibull
  geom_line(aes(x = t_predict, y = surv_weib_comb_con, color = "Control - Weibull"),
linewidth = 1) +
  geom_line(aes(x = t_predict, y = surv_weib_comb_exp, color = "Experimental - Weibull"),
linetype = "dashed", linewidth = 1) +

  # Log-normal
  geom_line(aes(x = t_predict, y = surv_lnorm_comb_con, color = "Control - Log-normal"),
linewidth = 1) +
  geom_line(aes(x = t_predict, y = surv_lnorm_comb_exp, color = "Experimental - Log-
normal"), linetype = "dashed", linewidth = 1) +

  # Log-logistic
  geom_line(aes(x = t_predict, y = surv_logl_comb_con, color = "Control - Log-logistic"),
linewidth = 1) +
  geom_line(aes(x = t_predict, y = surv_logl_comb_exp, color = "Experimental - Log-
logistic"), linetype = "dashed", linewidth = 1) +

  # Gompertz
  geom_line(aes(x = t_predict, y = surv_gompertz_comb_con, color = "Control - Gompertz"),
linewidth = 1) +
  geom_line(aes(x = t_predict, y = surv_gompertz_comb_exp, color = "Experimental -
Gompertz"), linetype = "dashed", linewidth = 1) +

  # Gamma
  geom_line(aes(x = t_predict, y = surv_gamma_comb_con, color = "Control - Gamma"),
linewidth = 1) +
  geom_line(aes(x = t_predict, y = surv_gamma_comb_exp, color = "Experimental -
Gamma"), linetype = "dashed", linewidth = 1) +

  # Generalized Gamma
  geom_line(aes(x = t_predict, y = surv_GG_comb_con, color = "Control - Generalized
Gamma"), linewidth = 1) +
  geom_line(aes(x = t_predict, y = surv_GG_comb_exp, color = "Experimental - Generalized
Gamma"), linetype = "dashed", linewidth = 1) +

  # KM Curves
  geom_step(aes(x = km_surv_con$time, y = km_surv_con$surv, color = "Control - KM"),
linewidth = 1) +
  geom_step(aes(x = km_surv_exp$time, y = km_surv_exp$surv, color = "Experimental -
KM"), linewidth = 1) +

  scale_color_manual(values = c(
    "Control - KM" = "#377EB8", "Experimental - KM" = "#E41A1C",
    "Control - Exponential" = "black", "Experimental - Exponential" = "black",
    "Control - Weibull" = "blue", "Experimental - Weibull" = "blue",
    "Control - Log-normal" = "purple", "Experimental - Log-normal" = "purple",
    "Control - Log-logistic" = "darkgreen", "Experimental - Log-logistic" = "darkgreen",
    "Control - Gompertz" = "brown", "Experimental - Gompertz" = "brown",
    "Control - Gamma" = "orange", "Experimental - Gamma" = "orange",
    "Control - Generalized Gamma" = "deeppink", "Experimental - Generalized Gamma" =
"deeppink"
  )) +

  scale_fill_manual(values = c("Control - 95% CI" = "#377EB8", "Experimental - 95% CI" =
"#E41A1C"), guide = "none") +

  labs(title = "Parametric Models and KM Curves",
       x = "Time since randomisation (months)",
       y = "Proportion surviving",
       color = "Model") +

  scale_x_continuous(limits = c(0, 240), breaks = seq(0, 240, by = 10), expand = c(0, 0.05))
+
  scale_y_continuous(limits = c(0, 1), breaks = seq(0, 1, by = 0.1), expand = c(0.02, 0)) +

  theme_bw() +
  theme(legend.position = c(0.7, 0.7),
        text = element_text(size = 14),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank()) +
  guides(color = guide_legend(ncol = 2))

# Plot
Survival_extrapolations_all_models


#------------------------------------------------
# Survival overlay: KM + parametric (Weibull+Log-normal+GenGamma)
#------------------------------------------------

Survival_s1 <- ggplot() +

  # KM CI ribbons
  geom_ribbon(aes(x = km_surv_con$time, ymin = km_surv_con$lower, ymax =
km_surv_con$upper), fill = "#377EB8", alpha = 0.1) +
  geom_ribbon(aes(x = km_surv_exp$time, ymin = km_surv_exp$lower, ymax =
km_surv_exp$upper), fill = "#E41A1C", alpha = 0.1) +

  # KM curves
  geom_step(aes(x = km_surv_con$time, y = km_surv_con$surv, color = "Control - KM
(solid)", linetype = "Control - KM (solid)"), linewidth = 1.5) +
  geom_step(aes(x = km_surv_exp$time, y = km_surv_exp$surv, color = "Experimental - KM
(solid)", linetype = "Experimental - KM (solid)"), linewidth = 1.5) +

  # Weibull
  geom_line(aes(x = t_predict, y = surv_weibull_ind_con, color = "Control - Weibull
(dashed)", linetype = "Control - Weibull (dashed)"), linewidth = 1.2) +
  geom_line(aes(x = t_predict, y = surv_weibull_ind_exp, color = "Experimental - Weibull
(dashed)", linetype = "Experimental - Weibull (dashed)"), linewidth = 1.2) +

  # Log-normal
  geom_line(aes(x = t_predict, y = surv_lnorm_ind_con, color = "Control - Log-normal
(dashed)", linetype = "Control - Log-normal (dashed)"), linewidth = 1.2) +
  geom_line(aes(x = t_predict, y = surv_lnorm_ind_exp, color = "Experimental - Log-normal
(dashed)", linetype = "Experimental - Log-normal (dashed)"), linewidth = 1.2) +

  # GenGamma
  geom_line(aes(x = t_predict, y = surv_GG_ind_con, color = "Control - GenGamma
(dashed)", linetype = "Control - GenGamma (dashed)"), linewidth = 1.2) +
  geom_line(aes(x = t_predict, y = surv_GG_ind_exp, color = "Experimental - GenGamma
(dashed)", linetype = "Experimental - GenGamma (dashed)"), linewidth = 1.2) +

  # Color mapping
  scale_color_manual(values = c(
    "Control - KM (solid)" = "#377EB8",
    "Experimental - KM (solid)" = "#E41A1C",
    "Control - Weibull (dashed)" = "#ff7f00",
    "Experimental - Weibull (dashed)" = "#ff7f50",
    "Control - Log-normal (dashed)" = "black",
    "Experimental - Log-normal (dashed)" = "black",
    "Control - GenGamma (dashed)" = "#33a02c",
    "Experimental - GenGamma (dashed)" = "#33a02c"
  )) +

  # Linetype mapping
  scale_linetype_manual(values = c(
    "Control - KM (solid)" = "solid",
    "Experimental - KM (solid)" = "solid",
    "Control - Weibull (dashed)" = "dashed",
    "Experimental - Weibull (dashed)" = "dashed",
    "Control - Log-normal (dashed)" = "dashed",
    "Experimental - Log-normal (dashed)" = "dashed",
    "Control - GenGamma (dashed)" = "dashed",
    "Experimental - GenGamma (dashed)" = "dashed"
  )) +

  # Labels
  labs(title = "Parametric model extrapolation: independent fits (short term)",
       subtitle = "Weibull, Log-normal, Generalised Gamma vs KM",
       x = "Time since randomisation (months)",
       y = "Proportion surviving",
       color = "Line (Group - Model - Type)",
       linetype = "Line (Group - Model - Type)") +

  # Axes →
  scale_x_continuous(limits = c(0, 40), breaks = seq(0, 40, by = 5)) +
  scale_y_continuous(limits = c(0, 1), breaks = seq(0, 1, by = 0.35)) +

  # Theme settings
  theme_bw() +
  theme(legend.position = c(3, 4),
        text = element_text(size = 12),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank()) +

  # Legend guide
  guides(color = guide_legend(ncol = 2),
         linetype = guide_legend(ncol = 2))

# Display the plot
Survival_s1
```

```
# Extrapolate to 240 months
Survival_etra_s1 <- Survival_s1 +
  scale_x_continuous(limits = c(0, 240), breaks = seq(0, 240, by = 30), expand = c(0, 0.05))

Survival_etra_s1

#---------------------------------------------------------------------------------------------------- #
#******  Construct hazard plots ****************************************
#---------------------------------------------------------------------------------------------------- #
Hazard_s1 <- ggplot() +

  # KM smoothed hazard → Control / Experimental
  geom_step(aes(x = hazard0_bshazard$time, y = hazard0_bshazard$hazard, color =
"Control - KM (solid)", linetype = "Control - KM (solid)"), linewidth = 1.2) +
  geom_step(aes(x = hazard1_bshazard$time, y = hazard1_bshazard$hazard, color =
"Experimental - KM (solid)", linetype = "Experimental - KM (solid)"), linewidth = 1.2) +

  # Weibull hazard
  geom_line(aes(x = t_predict, y = haz_weibull_ind_con, color = "Control - Weibull (dashed)",
linetype = "Control - Weibull (dashed)"), linewidth = 1) +
  geom_line(aes(x = t_predict, y = haz_weibull_ind_exp, color = "Experimental - Weibull
(dashed)", linetype = "Experimental - Weibull (dashed)"), linewidth = 1) +

  # Log-normal hazard
  geom_line(aes(x = t_predict, y = haz_lnorm_ind_con, color = "Control - Log-normal
(dashed)", linetype = "Control - Log-normal (dashed)"), linewidth = 1) +
  geom_line(aes(x = t_predict, y = haz_lnorm_ind_exp, color = "Experimental - Log-normal
(dashed)", linetype = "Experimental - Log-normal (dashed)"), linewidth = 1) +

  # GenGamma hazard
  geom_line(aes(x = t_predict, y = haz_GG_ind_con, color = "Control - GenGamma
(dashed)", linetype = "Control - GenGamma (dashed)"), linewidth = 1) +
  geom_line(aes(x = t_predict, y = haz_GG_ind_exp, color = "Experimental - GenGamma
(dashed)", linetype = "Experimental - GenGamma (dashed)"), linewidth = 1) +

  # Color mapping
  scale_color_manual(values = c(
    "Control - KM (solid)" = "#377EB8",
    "Experimental - KM (solid)" = "#E41A1C",
    "Control - Weibull (dashed)" = "#ff7f00",
    "Experimental - Weibull (dashed)" = "#ff7f50",
    "Control - Log-normal (dashed)" = "black",
    "Experimental - Log-normal (dashed)" = "grey",
    "Control - GenGamma (dashed)" = "#33a02c",
    "Experimental - GenGamma (dashed)" = "#b2df8a"
  )) +
  # Linetype mapping
  scale_linetype_manual(values = c(
    "Control - KM (solid)" = "solid",
    "Experimental - KM (solid)" = "solid",
    "Control - Weibull (dashed)" = "dashed",
    "Experimental - Weibull (dashed)" = "dashed",
    "Control - Log-normal (dashed)" = "dashed",
    "Experimental - Log-normal (dashed)" = "dashed",
    "Control - GenGamma (dashed)" = "dashed",
    "Experimental - GenGamma (dashed)" = "dashed"
  )) +

  # Labels
  labs(title = "Hazard functions: independent fits",
       subtitle = "Weibull, Log-normal, Generalised Gamma vs KM smoothed hazard",
       x = "Time since randomisation (months)",
       y = "Hazard per person-month",
       color = "Line (Group - Model - Type)",
       linetype = "Line (Group - Model - Type)") +

  # Axes
  scale_x_continuous(limits = c(0, 240), breaks = seq(0, 240, by = 20)) +
  scale_y_continuous(limits = c(0, NA), expand = c(0, 0)) +

  # Theme settings
  theme_bw() +
  theme(legend.position = c(0.7, 0.5),
        text = element_text(size = 10),
        panel.grid.major.y = element_blank(),
        panel.grid.minor.y = element_blank()) +

  # Legend guide
  guides(color = guide_legend(ncol = 2),
         linetype = guide_legend(ncol = 2))


#----------------------------------------------------------------------------------------------------- #
#******  Construct plots of the implied treatment effect over time
*****************************************************
#----------------------------------------------------------------------------------------------------- #
# Calculate HR = hazard_treatment / hazard_control
HR_weibull <- haz_weibull_ind_exp / haz_weibull_ind_con
HR_lognorm <- haz_lnorm_ind_exp / haz_lnorm_ind_con

# Clean infinite or NA values (e.g. divide by zero)
HR_weibull[!is.finite(HR_weibull)] <- NA
```

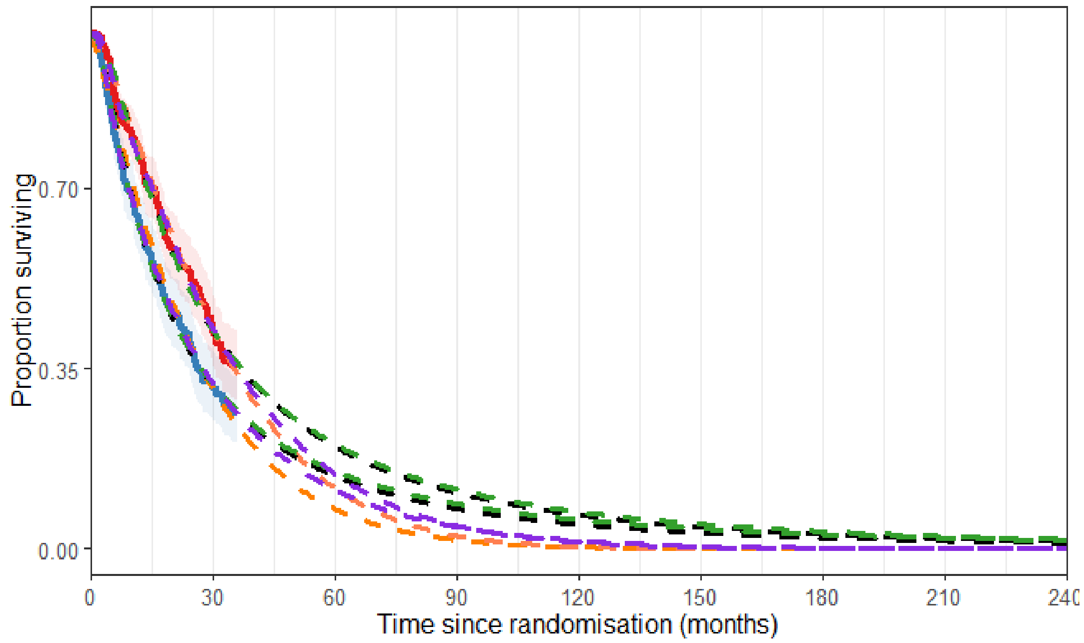
```
HR_lognorm[!is.finite(HR_lognorm)] <- NA

Implied_HR_plot_s1 <- ggplot() +

  geom_line(aes(x = t_predict, y = HR_weibull, color = "Weibull", linetype = "Weibull"),
linewidth = 1.2) +
  geom_line(aes(x = t_predict, y = HR_lognorm, color = "Log-normal", linetype = "Log-
normal"), linewidth = 1.2) +

  geom_hline(yintercept = 1.0, color = "black", linetype = "dashed", linewidth = 0.8) +

  scale_x_continuous(limits = c(0, 40), breaks = seq(0, 40, by = 5)) +
  scale_y_continuous(limits = c(0, NA), breaks = seq(0, 2, by = 0.2), expand = c(0, 0)) +

  scale_color_manual(values = c("Weibull" = "#ff7f00", "Log-normal" = "#6a3d9a")) +
  scale_linetype_manual(values = c("Weibull" = "dashed", "Log-normal" = "dotdash")) +

  labs(title = "Implied Hazard Ratio over Time (short term)",
      subtitle = "Derived from independent models",
      x = "Time since randomisation (months)",
      y = "Implied HR (experimental / control)",
      color = "Model",
      linetype = "Model") +

  theme_bw() +
  theme(legend.position = c(0.75, 0.3),
       text = element_text(size = 12),
       panel.grid.minor = element_blank())

Implied_HR_plot_s1_extra <- Implied_HR_plot_s1 +
  scale_x_continuous(limits = c(0, 240), breaks = seq(0, 240, by = 20))

#------------------------------------------------------------------------------------------------------- #
#******  1.5 Compare base model and selected parametric model
******************************************
#------------------------------------------------------------------------------------------------------- #

#****** Survival Overlay Plot（Log-normal independent vs Exponential combined）
******************************************
Survival_log_vs_exp <- ggplot() +
  # KM confidence bands
  geom_ribbon(aes(x = km_surv_con$time, ymin = km_surv_con$lower, ymax =
km_surv_con$upper), fill = "#377EB8", alpha = 0.1) +
  geom_ribbon(aes(x = km_surv_exp$time, ymin = km_surv_exp$lower, ymax =
km_surv_exp$upper), fill = "#E41A1C", alpha = 0.1) +

  # KM curves
  geom_step(aes(x = km_surv_con$time, y = km_surv_con$surv, color = "Control - KM", linetype = "Control - KM"), linewidth = 1.4) +
  geom_step(aes(x = km_surv_exp$time, y = km_surv_exp$surv, color = "Experimental - KM", linetype = "Experimental - KM"), linewidth = 1.4) +

  # Exponential combined
  geom_line(aes(x = t_predict, y = surv_exp_comb_con, color = "Control - Exponential", linetype = "Control - Exponential"), linewidth = 1.1) +
  geom_line(aes(x = t_predict, y = surv_exp_comb_exp, color = "Experimental - Exponential", linetype = "Experimental - Exponential"), linewidth = 1.1) +

  # Log-normal independent
  geom_line(aes(x = t_predict, y = surv_lnorm_ind_con, color = "Control - Log-normal", linetype = "Control - Log-normal"), linewidth = 1.1) +
  geom_line(aes(x = t_predict, y = surv_lnorm_ind_exp, color = "Experimental - Log-normal", linetype = "Experimental - Log-normal"), linewidth = 1.1) +

  scale_color_manual(values = c(
    "Control - KM" = "#377EB8",
    "Experimental - KM" = "#E41A1C",
    "Control - Exponential" = "#33a02c",
    "Experimental - Exponential" = "#b2df8a",
    "Control - Log-normal" = "black",
    "Experimental - Log-normal" = "black"
  )) +
  scale_linetype_manual(values = c(
    "Control - KM" = "solid",
    "Experimental - KM" = "solid",
    "Control - Exponential" = "dashed",
    "Experimental - Exponential" = "dashed",
    "Control - Log-normal" = "dashed",
    "Experimental - Log-normal" = "dashed"
  )) +
  labs(
    title = "Survival Curves: Exponential (combined) vs Log-normal (independent)",
    subtitle = "Compared against Kaplan-Meier estimates",
    x = "Time since randomisation (months)",
    y = "Proportion surviving",
    color = "Group - Model",
    linetype = "Group - Model"
  ) +
  scale_x_continuous(limits = c(0, 40), breaks = seq(0, 40, 5)) +
  scale_y_continuous(limits = c(0, 1), breaks = seq(0, 1, 0.25)) +
  theme_bw() +
  theme(legend.position = c(20, 0.25),
        text = element_text(size = 11),
        panel.grid.major.y = element_blank(),
    panel.grid.minor.y = element_blank())

Survival_log_vs_exp_extended <- Survival_log_vs_exp +
  scale_x_continuous(limits = c(0, 240), breaks = seq(0, 240, 30), expand = c(0, 0.02))

#****** Hazard Plot（Log-normal independent vs Exponential combined）
*******************************************
Hazard_log_vs_exp <- ggplot() +
  # KM smoothed hazard
  geom_step(aes(x = hazard0_bshazard$time, y = hazard0_bshazard$hazard, color =
"Control - KM", linetype = "Control - KM"), linewidth = 1.2) +
  geom_step(aes(x = hazard1_bshazard$time, y = hazard1_bshazard$hazard, color =
"Experimental - KM", linetype = "Experimental - KM"), linewidth = 1.2) +

  # Exponential combined
  geom_line(aes(x = t_predict, y = haz_exp_comb_con, color = "Control - Exponential",
linetype = "Control - Exponential"), linewidth = 1) +
  geom_line(aes(x = t_predict, y = haz_exp_comb_exp, color = "Experimental - Exponential",
linetype = "Experimental - Exponential"), linewidth = 1) +

  # Log-normal independent
  geom_line(aes(x = t_predict, y = haz_lnorm_ind_con, color = "Control - Log-normal",
linetype = "Control - Log-normal"), linewidth = 1) +
  geom_line(aes(x = t_predict, y = haz_lnorm_ind_exp, color = "Experimental - Log-normal",
linetype = "Experimental - Log-normal"), linewidth = 1) +

  scale_color_manual(values = c(
    "Control - KM" = "#377EB8",
    "Experimental - KM" = "#E41A1C",
    "Control - Exponential" = "#33a02c",
    "Experimental - Exponential" = "#b2df8a",
    "Control - Log-normal" = "black",
    "Experimental - Log-normal" = "grey40"
  )) +
  scale_linetype_manual(values = c(
    "Control - KM" = "solid",
    "Experimental - KM" = "solid",
    "Control - Exponential" = "dashed",
    "Experimental - Exponential" = "dashed",
    "Control - Log-normal" = "dashed",
    "Experimental - Log-normal" = "dashed"
  )) +
  labs(
    title = "Hazard Functions: Exponential (combined) vs Log-normal (independent)",
    subtitle = "Compared to KM smoothed hazard",
    x = "Time since randomisation (months)",
    y = "Hazard per person-month",
    color = "Group - Model",
    linetype = "Group - Model"
  ) +
  scale_x_continuous(limits = c(0, 240), breaks = seq(0, 240, 30)) +
  scale_y_continuous(limits = c(0, NA), expand = c(0, 0)) +
  theme_bw() +
  theme(legend.position = c(0.75, 0.6),
        text = element_text(size = 10),
        panel.grid.minor.y = element_blank())

#****** Hazard Plot（Log-normal independent vs Exponential combined）
******************************************
HR_lognorm_vs_exp <- haz_lnorm_ind_exp / haz_exp_comb_exp
HR_lognorm_vs_exp[!is.finite(HR_lognorm_vs_exp)] <- NA

Implied_HR_log_vs_exp <- ggplot() +
  geom_line(aes(x = t_predict, y = HR_lognorm_vs_exp, color = "Log-normal vs Exp",
linetype = "Log-normal vs Exp"), linewidth = 1.2) +
  geom_hline(yintercept = 1.0, color = "black", linetype = "dashed", linewidth = 0.8) +

  scale_x_continuous(limits = c(0, 40), breaks = seq(0, 40, 5)) +
  scale_y_continuous(limits = c(0, NA), breaks = seq(0, 2, by = 0.2)) +
  scale_color_manual(values = c("Log-normal vs Exp" = "#984ea3")) +
  scale_linetype_manual(values = c("Log-normal vs Exp" = "dotdash")) +

  labs(
    title = "Implied Hazard Ratio over Time",
    subtitle = "Log-normal (independent) vs Exponential (combined)",
    x = "Time since randomisation (months)",
    y = "Implied HR (lognorm / exp)",
    color = "Model Comparison",
    linetype = "Model Comparison"
  ) +
  theme_bw() +
  theme(legend.position = c(0.7, 0.25),
        text = element_text(size = 12),
        panel.grid.minor = element_blank())

Implied_HR_log_vs_exp_ext <- Implied_HR_log_vs_exp +
  scale_x_continuous(limits = c(0, 240), breaks = seq(0, 240, 30))
```

Appendix 2: Flexible Parametric Model

```
#----------------------------------------------------------------------------------------------------------------------#
#------- Bonus -------------------#
#----------------------------------------------------------------------------------------------------------------------#


#----------------------------------------------------------------------------------------------------------------------#
#------- Flexible Parametric Model: Combined (PH) and Independent (Non-PH) for PFS ----#
#----------------------------------------------------------------------------------------------------------------------#

# Combined model (proportional hazards assumption)
fit_fpm_pfs <- stpm2(Surv(time = PFS_months, event = event_num) ~ trtgrp_num,
                     data = data, df = 4, scale = "hazard")
fit_fpm_pfs

# AIC / BIC
AIC(fit_fpm_pfs)
BIC(fit_fpm_pfs)

# RMST truncated at 20 months
rmst_fpm_pfs_con <- integrate(function(t) {
  predict(fit_fpm_pfs, newdata = data.frame(PFS_months = t, trtgrp_num = 0), type = "surv")
}, lower = 0, upper = 20)$value

rmst_fpm_pfs_exp <- integrate(function(t) {
  predict(fit_fpm_pfs, newdata = data.frame(PFS_months = t, trtgrp_num = 1), type = "surv")
}, lower = 0, upper = 20)$value

# Survival & hazard predictions for extrapolation
surv_fpm_comb_con_pfs <- predict(fit_fpm_pfs, newdata = data.frame(PFS_months =
t_predict, trtgrp_num = 0), type = "surv")
surv_fpm_comb_exp_pfs <- predict(fit_fpm_pfs, newdata = data.frame(PFS_months =
t_predict, trtgrp_num = 1), type = "surv")

haz_fpm_comb_con_pfs <- predict(fit_fpm_pfs, newdata = data.frame(PFS_months =
t_predict, trtgrp_num = 0), type = "hazard")
haz_fpm_comb_exp_pfs <- predict(fit_fpm_pfs, newdata = data.frame(PFS_months =
t_predict, trtgrp_num = 1), type = "hazard")


# Independent models (non-proportional hazards)
fit_fpm2_con_pfs <- stpm2(Surv(time = PFS_months, event = event_num) ~ 1,
                          data = subset(data, trtgrp_num == 0), df = 4, scale = "hazard")
fit_fpm2_exp_pfs <- stpm2(Surv(time = PFS_months, event = event_num) ~ 1,
                          data = subset(data, trtgrp_num == 1), df = 4, scale = "hazard")
```

```
# AIC / BIC
AIC(fit_fpm2_con_pfs); BIC(fit_fpm2_con_pfs)
AIC(fit_fpm2_exp_pfs); BIC(fit_fpm2_exp_pfs)

# RMST (truncated at 20 months)
rmst_fpm2_con_pfs <- integrate(function(t) {
  predict(fit_fpm2_con_pfs, newdata = data.frame(PFS_months = t), type = "surv")
}, lower = 0, upper = 20)$value

rmst_fpm2_exp_pfs <- integrate(function(t) {
  predict(fit_fpm2_exp_pfs, newdata = data.frame(PFS_months = t), type = "surv")
}, lower = 0, upper = 20)$value

# Survival & hazard predictions
surv_fpm_ind_con_pfs <- predict(fit_fpm2_con_pfs, newdata = data.frame(PFS_months =
t_predict), type = "surv")
surv_fpm_ind_exp_pfs <- predict(fit_fpm2_exp_pfs, newdata = data.frame(PFS_months =
t_predict), type = "surv")

haz_fpm_ind_con_pfs <- predict(fit_fpm2_con_pfs, newdata = data.frame(PFS_months =
t_predict), type = "hazard")
haz_fpm_ind_exp_pfs <- predict(fit_fpm2_exp_pfs, newdata = data.frame(PFS_months =
t_predict), type = "hazard")



# Add: Flexible parametric model (independent)
Survival_s1_bonus <- Survival_etra_s1 +
  geom_line(aes(x = t_predict, y = surv_fpm_ind_con_pfs, color = "Control - FPM (dashed)",
linetype = "Control - FPM (dashed)"), linewidth = 1.2) +
  geom_line(aes(x = t_predict, y = surv_fpm_ind_exp_pfs, color = "Experimental - FPM
(dashed)", linetype = "Experimental - FPM (dashed)"), linewidth = 1.2) +

  scale_color_manual(values = c(
    # Existing
    "Control - KM (solid)" = "#377EB8",
    "Experimental - KM (solid)" = "#E41A1C",
    "Control - Weibull (dashed)" = "#ff7f00",
    "Experimental - Weibull (dashed)" = "#ff7f50",
    "Control - Log-normal (dashed)" = "black",
    "Experimental - Log-normal (dashed)" = "black",
    "Control - GenGamma (dashed)" = "#33a02c",
    "Experimental - GenGamma (dashed)" = "#33a02c",
    # NEW
    "Control - FPM (dashed)" = "#8A2BE2",
    "Experimental - FPM (dashed)" = "#8A2BE2"
  )) +
  scale_linetype_manual(values = c(
    # Existing
    "Control - KM (solid)" = "solid",
    "Experimental - KM (solid)" = "solid",
    "Control - Weibull (dashed)" = "dashed",
    "Experimental - Weibull (dashed)" = "dashed",
    "Control - Log-normal (dashed)" = "dashed",
    "Experimental - Log-normal (dashed)" = "dashed",
    "Control - GenGamma (dashed)" = "dashed",
    "Experimental - GenGamma (dashed)" = "dashed",
    # NEW
    "Control - FPM (dashed)" = "dashed",
    "Experimental - FPM (dashed)" = "dashed"
  ))


# Add: Flexible parametric model (independent)
Hazard_s1 <- Hazard_s1 +
  geom_line(aes(x = t_predict, y = haz_fpm_ind_con_pfs, color = "Control - FPM (dashed)",
linetype = "Control - FPM (dashed)"), linewidth = 1) +
  geom_line(aes(x = t_predict, y = haz_fpm_ind_exp_pfs, color = "Experimental - FPM
(dashed)", linetype = "Experimental - FPM (dashed)"), linewidth = 1) +

  scale_color_manual(values = c(
    # Existing
    "Control - KM (solid)" = "#377EB8",
    "Experimental - KM (solid)" = "#E41A1C",
    "Control - Weibull (dashed)" = "#ff7f00",
    "Experimental - Weibull (dashed)" = "#ff7f50",
    "Control - Log-normal (dashed)" = "black",
    "Experimental - Log-normal (dashed)" = "grey",
    "Control - GenGamma (dashed)" = "#33a02c",
    "Experimental - GenGamma (dashed)" = "#b2df8a",
    # NEW
    "Control - FPM (dashed)" = "#8A2BE2",
    "Experimental - FPM (dashed)" = "#8A2BE2"
  )) +

  scale_linetype_manual(values = c(
    # Existing
    "Control - KM (solid)" = "solid",
    "Experimental - KM (solid)" = "solid",
    "Control - Weibull (dashed)" = "dashed",
    "Experimental - Weibull (dashed)" = "dashed",
    "Control - Log-normal (dashed)" = "dashed",
    "Experimental - Log-normal (dashed)" = "dashed",
    "Control - GenGamma (dashed)" = "dashed",
    "Experimental - GenGamma (dashed)" = "dashed",
    # NEW
    "Control - FPM (dashed)" = "dashed",
    "Experimental - FPM (dashed)" = "dashed"
  ))
```

Figure 1: Survival Curves from Independent Parametric and Flexible Models



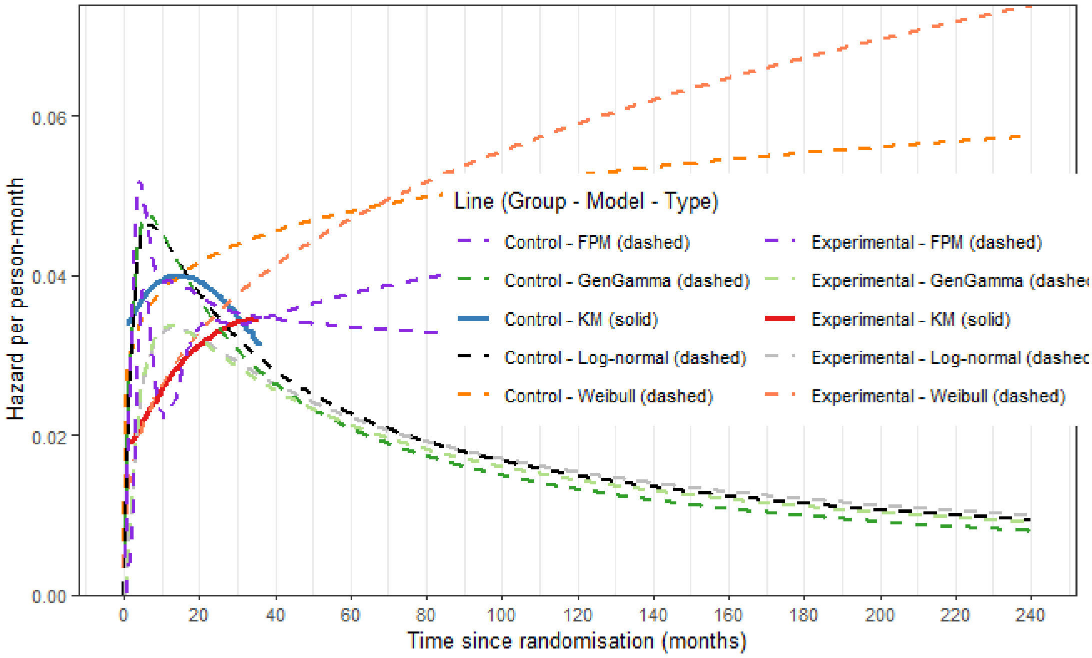
This plot compares progression-free survival (PFS) extrapolations using several independently fitted models, including:

- **Kaplan-Meier (KM)** curves for the control (blue solid line) and experimental group (red solid line)
- **Parametric models**: Weibull (orange), Log-normal (black), and Generalised Gamma (green)
- **Flexible Parametric Model (FPM)**: shown in purple dashed lines for both groups

Interpretation:

- The **Log-normal** and **GenGamma** models provide reasonable fits in the short term and align well with the KM curves.
- The **Weibull** model appears to underestimate long-term survival, with curves dropping too steeply.
- The **FPM** captures the overall shape of the KM data and allows more flexible long-term extrapolation, although its upper tail tends to predict slightly higher survival.

Figure 2: Hazard Functions from Independent Models Compared to KM Estimates



This figure displays time-varying hazard functions derived from the same set of models:

- **KM smoothed hazard** estimates (blue and red solid lines)
- **Weibull (orange)**: shows a continuously increasing hazard, which may not reflect the observed data
- **Log-normal (black)**: captures a rising and falling hazard trend
- **GenGamma (green)**: fits the overall hazard shape well with greater flexibility
- **FPM (purple)**: offers a smooth, flexible hazard trajectory, closely following the KM pattern, especially between 10 and 30 months

Interpretation:

- The **Weibull** model's monotonic increase in hazard may oversimplify the actual risk dynamics.
- **FPM** provides superior flexibility in modeling non-proportional and non-monotonic hazards, making it well-suited for realistic clinical scenarios where the risk of progression may rise and fall over time.