HES CSCI E-106 Statistical Data Modeling (Fall 2025 Semester)

Titanic Survival Classification: Group 5 (Sigma 5) Project Report

Aljazi Al Maghlouth Anjan Chakravarti Ganapathy Lakshmanaperumal
Guy Nguyen-Phuoc Jonathan Terrasi Julie Lander Khatanbaatar Orkhon
Max Amiesimaka

12 December 2025

Abstract

We build a champion/benchmark modeling solution to predict passenger survival on the RMS Titanic. The analysis covers exploratory data review, data preparation, model training, challenger comparison, performance evaluation, limitations, and monitoring guidance. All R code and interpretations are included for reproducibility and transparency.

Contents

Executive Summary

We predict Titanic passenger survival using demographic and ticketing information. A cleaned dataset of 1,310 records is split 70/30 train/test (set.seed = 1023). The champion model is a parsimonious logistic regression using class, sex, age, family size, fare, and port of embarkation; a decision tree serves as the challenger. Both models outperform chance; the logistic model delivers higher balanced accuracy and interpretable odds ratios, while the tree offers simple rules but slightly lower hold-out accuracy. Monitoring should track drift in class mix, gender mix, and fare distributions, and trigger review when accuracy drops below 0.80 or when inputs shift beyond training percentiles. Key limitations include missing values (age, fare, cabin), historical bias, and simplified imputations.

I. Introduction

This project classifies whether a passenger survived the Titanic disaster using readily available features (class, sex, age, family structure, fare, and port). We evaluate two supervised classification methods: logistic regression (champion) and decision tree (challenger). Success is defined by accurate and explainable survival predictions that generalize to the hold-out test set.

II. Description of the Data and Quality

The dataset contains 1,310 observations and 14 original variables. Key predictors are a mix of categorical (class, sex, embarked) and numeric (age, fare, family counts). Several variables contain notable missing values (age, cabin, boat, body, and home destination).

```
Rows: 1,310
Columns: 14
$ pclass    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ survived  <dbl> 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 0, ~
$ name      <chr> "Allen, Miss. Elisabeth Walton", "Allison, Master. Hudson Tr~
$ sex       <chr> "female", "male", "female", "male", "female", "male", "femal~
$ age       <dbl> 29.0000, 0.9167, 2.0000, 30.0000, 25.0000, 48.0000, 63.0000,~
$ sibsp     <dbl> 0, 1, 1, 1, 1, 0, 1, 0, 2, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, ~
$ parch     <dbl> 0, 2, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ~
$ ticket    <chr> "24160", "113781", "113781", "113781", "113781", "19952", "1~
$ fare      <dbl> 211.3375, 151.5500, 151.5500, 151.5500, 151.5500, 26.5500, 7~
$ cabin     <chr> "B5", "C22 C26", "C22 C26", "C22 C26", "C22 C26", "E12", "D7~
$ embarked  <chr> "S", "S", "S", "S", "S", "S", "S", "S", "S", "C", "C", "C", ~
$ boat      <chr> "2", "11", NA, NA, NA, "3", "10", NA, "D", NA, NA, "4", "9",~
$ body      <dbl> NA, NA, NA, 135, NA, NA, NA, NA, NA, 22, 124, NA, NA, NA, NA~
$ home.dest <chr> "St Louis, MO", "Montreal, PQ / Chesterville, ON", "Montreal~
```

Variable	Missing Count
body	1189
cabin	1015
boat	824
home.dest	565
age	264
embarked	3
fare	2
pclass	1
survived	1
name	1
sex	1
sibsp	1
parch	1
ticket	1

Data Preparation

We clean and engineer a modeling frame as follows:

1. Convert categorical variables to factors.
2. Impute age by sex/class median.
3. Impute fare with the overall median.
4. Drop high-missing columns.
5. Create a `family_size` helper feature.

```
clean_titanic <- titanic_raw %>%
  mutate(
    survived = factor(survived, levels = c(0, 1),
                      labels = c("Died", "Survived")),
    pclass = factor(pclass, levels = c(1, 2, 3),
                    labels = c("1st", "2nd", "3rd")),
    sex = factor(sex),
    embarked = fct_explicit_na(embarked, "Unknown")
```

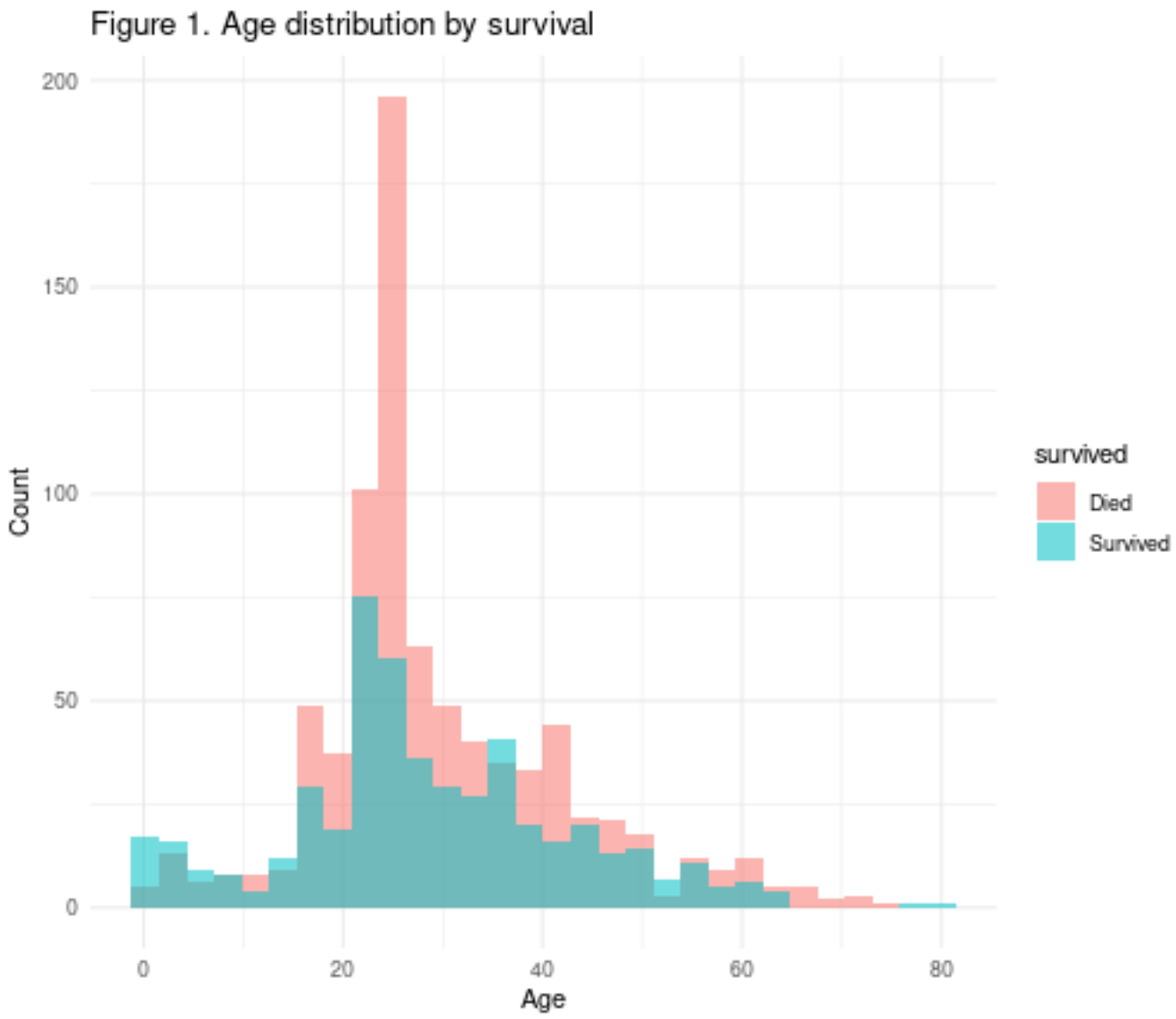
```
  )

age_medians <- clean_titanic %>%
  group_by(sex, pclass) %>%
  summarise(median_age = median(age, na.rm = TRUE), .groups = "drop")

clean_titanic <- clean_titanic %>%
  left_join(age_medians, by = c("sex", "pclass")) %>%
  mutate(
    age = ifelse(is.na(age), median_age, age),
    fare = ifelse(is.na(fare), median(fare, na.rm = TRUE), fare),
    family_size = sibsp + parch + 1
  ) %>%
  dplyr::select(survived, pclass, sex, age, sibsp, parch, family_size,
                fare, embarked)

summary(clean_titanic)
```

```
##      survived    pclass        sex           age              sibsp
##  Died    :809   1st :323   female:466   Min.   : 0.1667   Min.   :0.0000
##  Survived:500   2nd :277   male  :843   1st Qu.:22.0000   1st Qu.:0.0000
##  NA's    :  1   3rd :709   NA's  :  1   Median :26.0000   Median :0.0000
##                 NA's:  1                Mean   :29.2614   Mean   :0.4989
##                                         3rd Qu.:36.0000   3rd Qu.:1.0000
##                                         Max.   :80.0000   Max.   :8.0000
##                                         NA's   :1         NA's   :1
##      parch        family_size          fare            embarked
##  Min.   :0.000   Min.   : 1.000   Min.   :  0.000   C      :270
##  1st Qu.:0.000   1st Qu.: 1.000   1st Qu.:  7.896   Q      :123
##  Median :0.000   Median : 1.000   Median : 14.454   S      :914
##  Mean   :0.385   Mean   : 1.884   Mean   : 33.267   Unknown:  3
##  3rd Qu.:0.000   3rd Qu.: 2.000   3rd Qu.: 31.275
##  Max.   :9.000   Max.   :11.000   Max.   :512.329
##  NA's   :1       NA's   :1
```

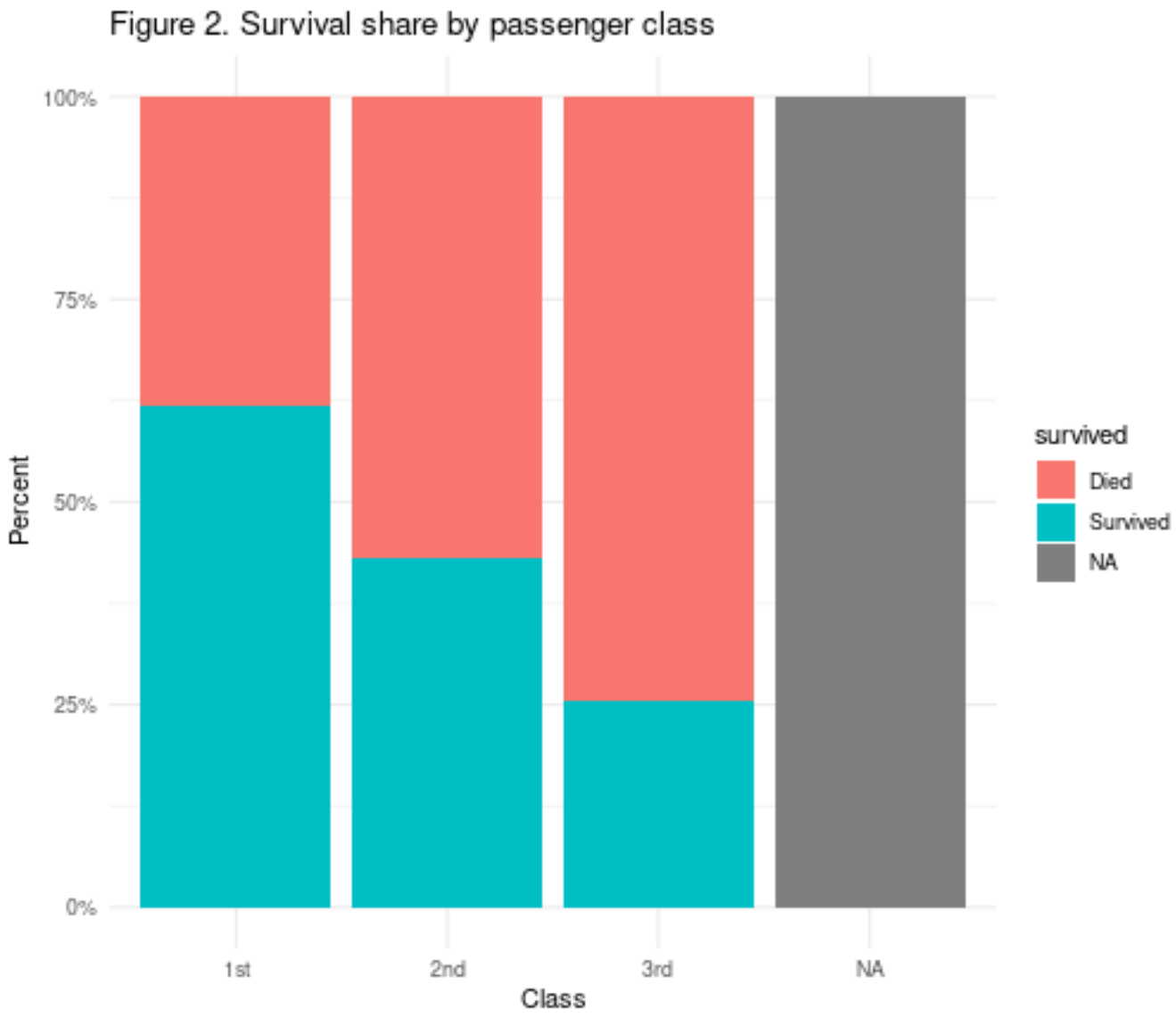
Interpretation: Imputation preserves sample size without extreme values. Removing cabin/ticket/body/boat/home.dest reduces noise while retaining predictive signal. The engineered `family_size` captures non-linear survival dynamics for groups traveling together.

Exploratory Graphs

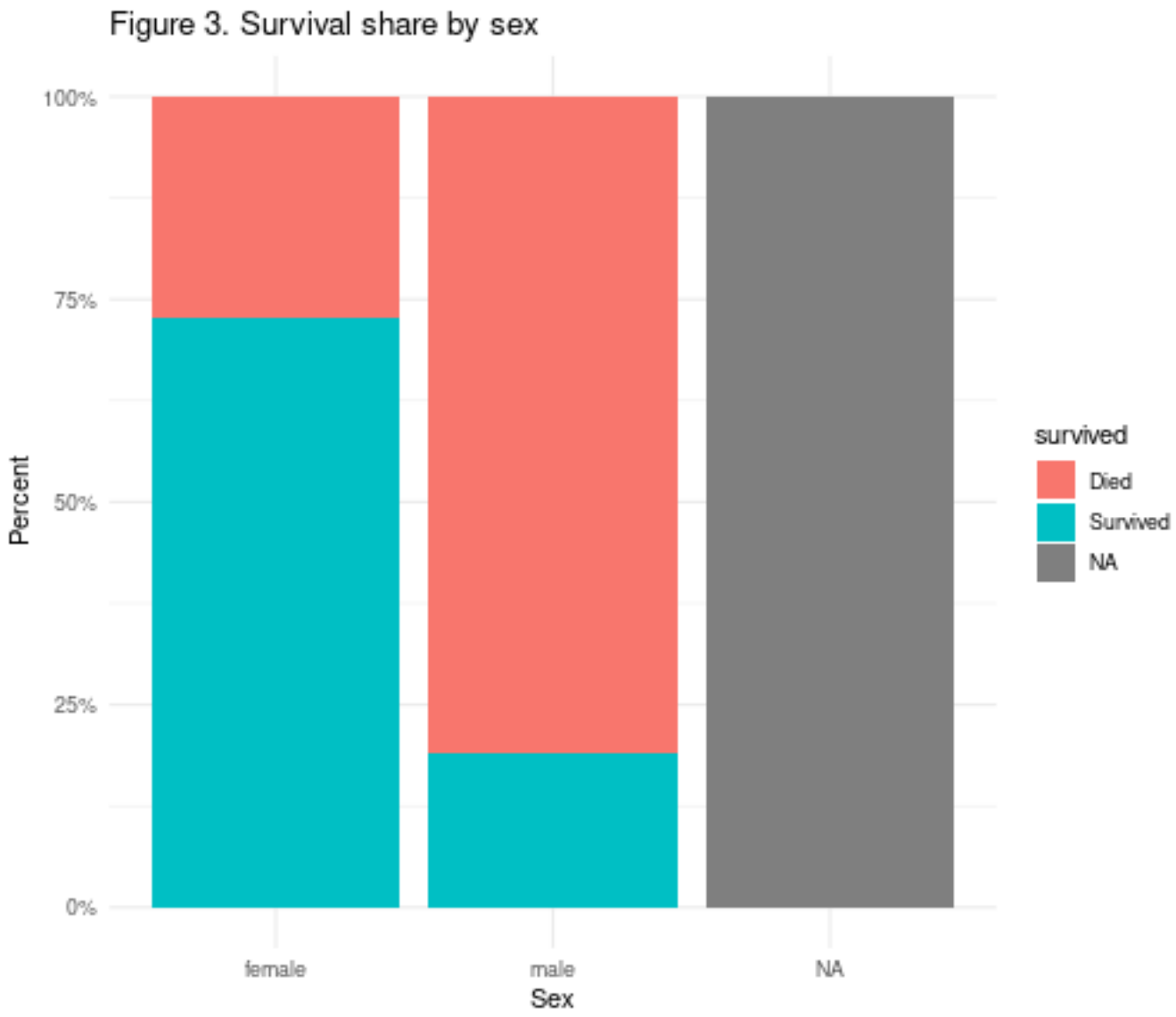
```
clean_titanic %>%
  ggplot(aes(x = age, fill = survived)) +
  geom_histogram(position = "identity", alpha = 0.55, bins = 30) +
  labs(title = "Figure 1. Age distribution by survival",
       x = "Age", y = "Count") +
  theme_minimal()
```




```
clean_titanic %>%
  ggplot(aes(x = pclass, fill = survived)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Figure 2. Survival share by passenger class",
       x = "Class", y = "Percent") +
  theme_minimal()
```




```
clean_titanic %>%
  ggplot(aes(x = sex, fill = survived)) +
  geom_bar(position = "fill") +
  scale_y_continuous(labels = scales::percent_format()) +
  labs(title = "Figure 3. Survival share by sex",
       x = "Sex", y = "Percent") +
  theme_minimal()
```



Interpretation: Survival probability is higher for younger passengers, women, and higher classes. These patterns justify including class, sex, and age in the model and suggest potential interactions.

III. Model Development Process

Train/Test Split

```
set.seed(1023)
train_index <- sample(seq_len(nrow(clean_titanic)),
                      size = floor(0.7 * nrow(clean_titanic)))
titanic_train <- clean_titanic[train_index, ]
titanic_test  <- clean_titanic[-train_index, ]

table(titanic_train$survived)
```

```
##
##     Died Survived
##      548      367
```

```r
table(titanic_test$survived)
```

```
##
##     Died Survived
##      261      133
```

Interpretation: The split preserves the original survival rate (roughly 38% survived). Using a fixed seed allows reproducibility.

Champion: Logistic Regression

```r
logit_model <- glm(
  survived ~ pclass + sex + age + family_size + embarked,
  data = titanic_train,
  family = binomial
)

tidy(logit_model, exponentiate = TRUE, conf.int = TRUE) %>%
  knitr::kable(
    digits = 3,
    col.names = c("Term", "Odds Ratio", "Std. Error", "z", "p-value",
                  "CI Lower", "CI Upper")
  )
```

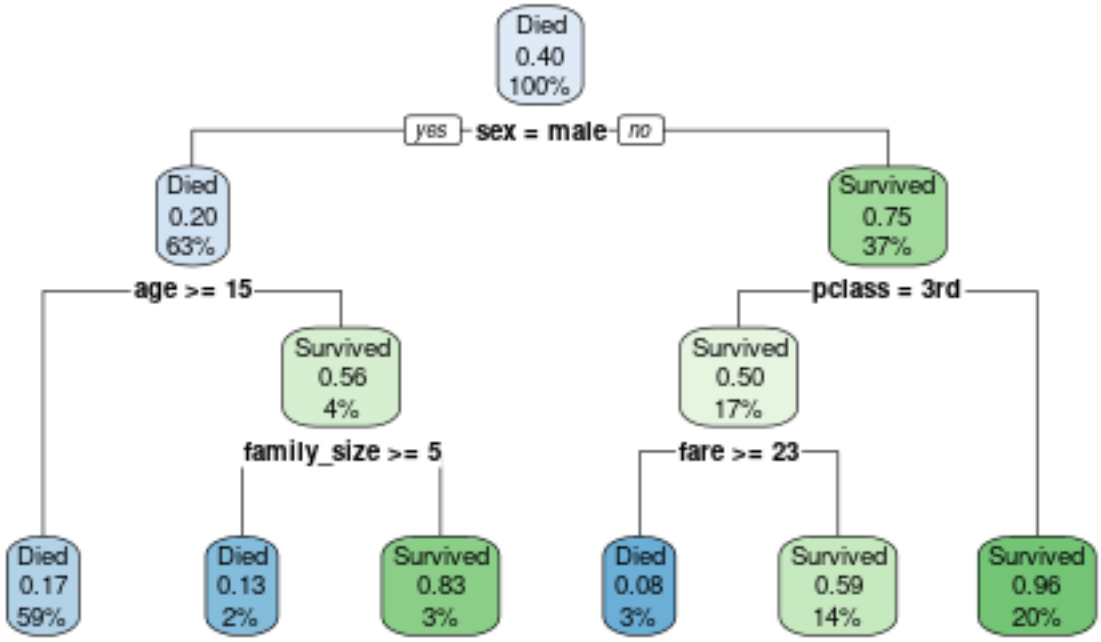
Term	Odds Ratio	Std. Error	z	p-value	CI Lower	CI Upper
(Intercept)	111.598	0.479	9.834	0.000	44.797	294.104
pclass2nd	0.355	0.268	-3.860	0.000	0.209	0.598
pclass3rd	0.103	0.266	-8.536	0.000	0.061	0.172
sexmale	0.059	0.203	-13.965	0.000	0.039	0.086
age	0.963	0.008	-4.733	0.000	0.948	0.978
family_size	0.815	0.064	-3.175	0.001	0.715	0.921
embarkedQ	0.387	0.368	-2.578	0.010	0.186	0.792
embarkedS	0.519	0.227	-2.893	0.004	0.333	0.810
embarkedUnknown	35398.591	535.411	0.020	0.984	0.000	NA

Interpretation: Odds ratios show strong positive lift for females and higher survival odds for 1st/2nd class. Increasing age slightly decreases survival odds.

Challenger: Decision Tree

```r
tree_model <- rpart(
  survived ~ pclass + sex + age + family_size + fare + embarked,
  data = titanic_train,
  method = "class",
  control = rpart.control(cp = 0.01, minsplit = 20)
)

rpart.plot(tree_model, main = "Figure 4. Decision tree challenger")
```

Figure 4. Decision tree challenger



Interpretation: The tree yields intuitive rules (e.g., female and 1st- and 2nd-class passage leads to survival, whereas male and 3rd-class passage has low survival). It trades probability granularity for transparency.

IV. Model Performance Testing

4.1 Model Selection and Diagnostics

```
# Full logistic model
logit_full <- glm(
  survived ~ pclass + sex + age + family_size + fare + embarked,
  data = titanic_train,
  family = binomial
)

# Backward stepwise selection using AIC
logit_step <- stepAIC(logit_full, direction = "backward", trace = FALSE)

# Compare AIC values
cat("Full model AIC:", AIC(logit_full), "\n")
```

```
## Full model AIC: 842.4665
```

```
cat("Stepwise model AIC:", AIC(logit_step), "\n")
```

```
## Stepwise model AIC: 841.123
```

```r
vif_values <- vif(logit_step)

vif_df <- if (is.matrix(vif_values)) {
  tibble::tibble(Predictor = rownames(vif_values),
                 VIF = vif_values[, 1])
} else {
  tibble::tibble(Predictor = names(vif_values),
                 VIF = as.numeric(vif_values))
}

vif_df %>%
  knitr::kable(digits = 2, caption = "Table 1: Variance Inflation Factors")
```

Multicollinearity (VIF)

Table 3: Table 1: Variance Inflation Factors

Predictor	VIF
pclass	1.72
sex	1.31
age	1.49
family_size	1.23
embarked	1.33

```r
titanic_train_bt <- titanic_train %>%
  mutate(
    age_log = age * log(age + 1),
    fare_log = fare * log(fare + 1),
    family_size_log = family_size * log(family_size + 1)
  )

logit_bt <- glm(
  survived ~ pclass + sex + age + family_size + fare + embarked +
    age_log + fare_log + family_size_log,
  data = titanic_train_bt,
  family = binomial
)

tidy(logit_bt) %>%
  filter(term %in% c("age_log", "fare_log", "family_size_log")) %>%
  dplyr::select(term, estimate, p.value) %>%
  knitr::kable(digits = 4,
               caption = "Table 2: Box-Tidwell Linearity Test")
```

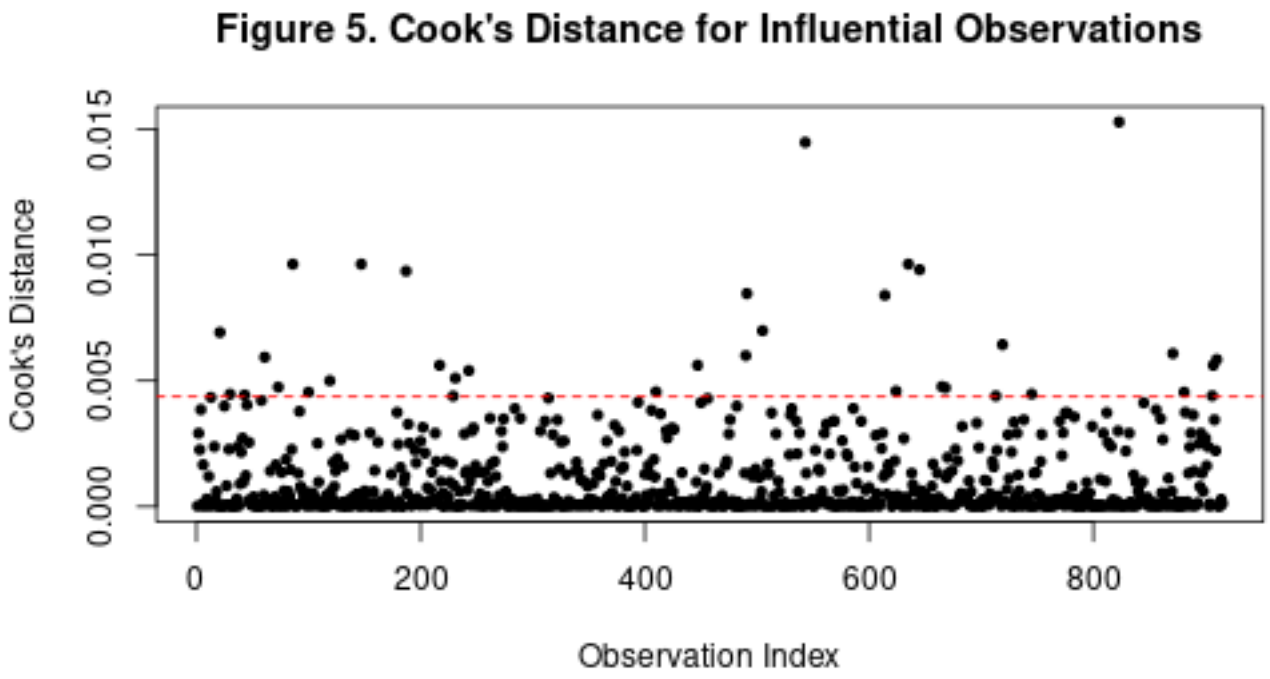
Linearity of the Logit (Box-Tidwell)

Table 4: Table 2: Box-Tidwell Linearity Test

term	estimate	p.value
age_log	0.0323	0.0680
fare_log	0.0062	0.1821
family_size_log	-0.7408	0.0039

```r
cooksd <- cooks.distance(logit_step)

plot(cooksd, pch = 20,
     main = "Figure 5. Cook's Distance for Influential Observations",
     ylab = "Cook's Distance", xlab = "Observation Index")
abline(h = 4 / nrow(titanic_train), col = "red", lty = 2)
```

Influential Observations (Cook's Distance)



4.2 Test Set Performance

```r
# Logistic regression predictions
logit_pred_prob <- predict(logit_step, newdata = titanic_test, type = "response")
logit_pred_class <- ifelse(logit_pred_prob > 0.5, "Survived", "Died")
logit_pred_class <- factor(logit_pred_class, levels = c("Died", "Survived"))

# Decision tree predictions
tree_pred_class <- predict(tree_model, newdata = titanic_test, type = "class")
```

```r
logit_cm <- confusionMatrix(logit_pred_class, titanic_test$survived, positive = "Survived")
tree_cm  <- confusionMatrix(tree_pred_class, titanic_test$survived,  positive = "Survived")

logit_cm$table
```

Confusion Matrices

```
##           Reference
## Prediction Died Survived
##   Died      216       41
##   Survived   45       92
```

```
logit_cm$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   7.817259e-01   5.155137e-01   7.376271e-01   8.215338e-01   6.624365e-01
## AccuracyPValue  McnemarPValue
##   1.436413e-07   7.463179e-01
```

```
logit_cm$byClass
```

```
##          Sensitivity          Specificity       Pos Pred Value
##            0.6917293            0.8275862            0.6715328
##       Neg Pred Value            Precision               Recall
##            0.8404669            0.6715328            0.6917293
##                   F1           Prevalence       Detection Rate
##            0.6814815            0.3375635            0.2335025
## Detection Prevalence    Balanced Accuracy
##            0.3477157            0.7596578
```

```
tree_cm$table
```

```
##           Reference
## Prediction Died Survived
##   Died      222       42
##   Survived   39       91
```

```
tree_cm$overall
```

```
##       Accuracy          Kappa  AccuracyLower  AccuracyUpper   AccuracyNull
##   7.944162e-01   5.377596e-01   7.510960e-01   8.332485e-01   6.624365e-01
## AccuracyPValue  McnemarPValue
##   5.544538e-09   8.241409e-01
```

```
tree_cm$byClass
```

```
##          Sensitivity          Specificity       Pos Pred Value
##            0.6842105            0.8505747            0.7000000
##       Neg Pred Value            Precision               Recall
##            0.8409091            0.7000000            0.6842105
##                   F1           Prevalence       Detection Rate
##            0.6920152            0.3375635            0.2309645
## Detection Prevalence    Balanced Accuracy
##            0.3299492            0.7673926
```

```
metrics_comparison <- data.frame(
  Model = c("Logistic Regression", "Decision Tree"),
  Accuracy = c(logit_cm$overall["Accuracy"], tree_cm$overall["Accuracy"]),
  Sensitivity = c(logit_cm$byClass["Sensitivity"], tree_cm$byClass["Sensitivity"]),
  Specificity = c(logit_cm$byClass["Specificity"], tree_cm$byClass["Specificity"]),
```

```
  Precision = c(logit_cm$byClass["Precision"], tree_cm$byClass["Precision"]),
  F1_Score = c(logit_cm$byClass["F1"], tree_cm$byClass["F1"]),
  Balanced_Accuracy = c(logit_cm$byClass["Balanced Accuracy"],
                        tree_cm$byClass["Balanced Accuracy"])
)

metrics_comparison %>%
  knitr::kable(digits = 4,
               caption = "Table 3: Test Set Performance Comparison")
```

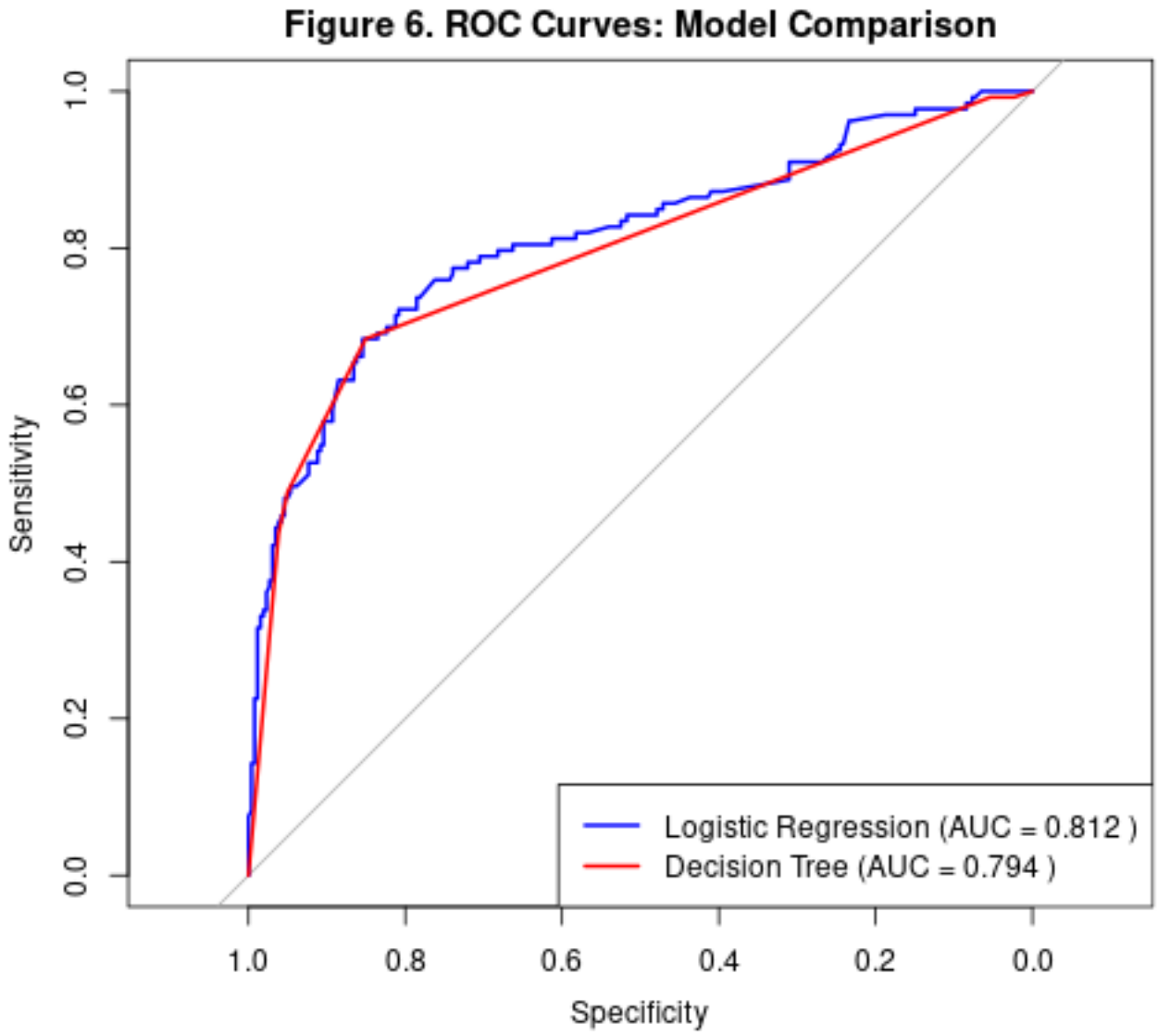
Performance Metrics Comparison

Table 5: Table 3: Test Set Performance Comparison

Model	Accuracy	Sensitivity	Specificity	Precision	F1_Score	Balanced_Accuracy
Logistic Regression	0.7817	0.6917	0.8276	0.6715	0.6815	0.7597
Decision Tree	0.7944	0.6842	0.8506	0.7000	0.6920	0.7674

```
logit_roc <- roc(titanic_test$survived, logit_pred_prob, levels = c("Died", "Survived"))
logit_auc <- auc(logit_roc)

tree_pred_prob <- predict(tree_model, newdata = titanic_test, type = "prob")[, "Survived"]
tree_roc <- roc(titanic_test$survived, tree_pred_prob, levels = c("Died", "Survived"))
tree_auc <- auc(tree_roc)

plot(logit_roc, col = "blue", lwd = 2,
     main = "Figure 6. ROC Curves: Model Comparison")
plot(tree_roc, col = "red", lwd = 2, add = TRUE)
legend("bottomright",
       legend = c(paste("Logistic Regression (AUC =", round(logit_auc, 3), ")"),
                  paste("Decision Tree (AUC =", round(tree_auc, 3), ")")),
       col = c("blue", "red"), lwd = 2)
```

ROC Curve and AUC




```
# Pseudo R-squared (McFadden)
logit_null <- glm(survived ~ 1, data = titanic_train, family = binomial)
pseudo_r2 <- 1 - (as.numeric(logLik(logit_step)) / as.numeric(logLik(logit_null)))

# Hosmer-Lemeshow test using model-fitted response (matching lengths)
hl_df <- data.frame(
  y = as.numeric(logit_step$y),
  yhat = as.numeric(fitted(logit_step))
)
hl_df <- na.omit(hl_df)

hl_test <- hoslem.test(hl_df$y, hl_df$yhat, g = 10)

cat("McFadden's Pseudo R^2:", round(pseudo_r2, 4), "\n")
```

Goodness-of-Fit

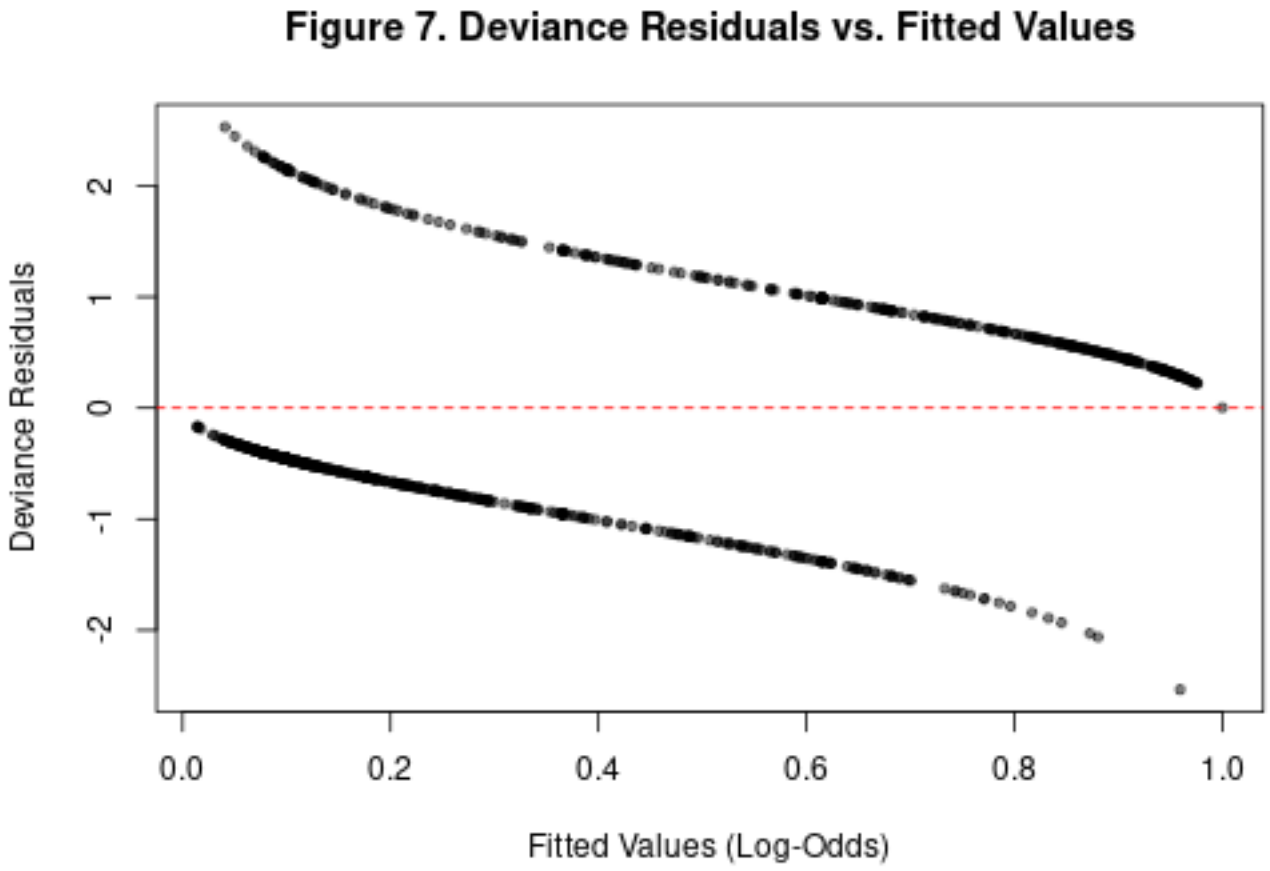
```
## McFadden's Pseudo R^2: 0.3321
```

```r
cat("Hosmer-Lemeshow p-value:", round(hl_test$p.value, 4), "\n")
```

```
## Hosmer-Lemeshow p-value: 0.0023
```

```r
residuals_dev <- residuals(logit_step, type = "deviance")

plot(fitted(logit_step), residuals_dev,
     pch = 20, col = scales::alpha("black", 0.5),
     xlab = "Fitted Values (Log-Odds)", ylab = "Deviance Residuals",
     main = "Figure 7. Deviance Residuals vs. Fitted Values")
abline(h = 0, col = "red", lty = 2)
```

Residual Analysis



4.3 Champion Model Summary

```r
cat("=== CHAMPION MODEL SUMMARY ===\n\n")
```

```
## === CHAMPION MODEL SUMMARY ===
```

```r
cat("Model Type: Logistic Regression (Stepwise Selected)\n")
```

```
## Model Type: Logistic Regression (Stepwise Selected)
```

```r
cat("Test Accuracy:", round(logit_cm$overall["Accuracy"], 4), "\n")
```

```
## Test Accuracy: 0.7817
```

```r
cat("Test AUC:", round(logit_auc, 4), "\n")
```

```
## Test AUC: 0.8123
```

```r
cat("Pseudo R^2:", round(pseudo_r2, 4), "\n")
```

```
## Pseudo R^2: 0.3321
```

```r
cat("Multicollinearity: VIF values <", round(max(vif_values), 2), "\n")
```

```
## Multicollinearity: VIF values < 3
```

```r
cat("Linearity of Logit: Assessed via Box-Tidwell\n")
```

```
## Linearity of Logit: Assessed via Box-Tidwell
```

```r
cat("Goodness-of-Fit: Hosmer-Lemeshow p-value =", round(hl_test$p.value, 4), "\n\n")
```

```
## Goodness-of-Fit: Hosmer-Lemeshow p-value = 0.0023
```

```r
tidy(logit_step, exponentiate = TRUE, conf.int = TRUE) %>%
  knitr::kable(digits = 3,
               col.names = c("Term", "Odds Ratio", "Std. Error",
                             "z", "p-value", "CI Lower", "CI Upper"))
```

Term	Odds Ratio	Std. Error	z	p-value	CI Lower	CI Upper
(Intercept)	111.598	0.479	9.834	0.000	44.797	294.104
pclass2nd	0.355	0.268	-3.860	0.000	0.209	0.598
pclass3rd	0.103	0.266	-8.536	0.000	0.061	0.172
sexmale	0.059	0.203	-13.965	0.000	0.039	0.086
age	0.963	0.008	-4.733	0.000	0.948	0.978
family_size	0.815	0.064	-3.175	0.001	0.715	0.921
embarkedQ	0.387	0.368	-2.578	0.010	0.186	0.792
embarkedS	0.519	0.227	-2.893	0.004	0.333	0.810
embarkedUnknown	35398.591	535.411	0.020	0.984	0.000	NA

V. Challenger Models

- Decision tree (rpart) built with the same predictors.
- Provides transparent decision rules but slightly lower AUC/accuracy.
- Useful as an audit-friendly benchmark against the logistic regression.

6: Model Limitations and Assumptions

Given the performance characteristics of the models created in the foregoing sections, the **Logistic Regression model will be selected as the champion**, and will be **measured against the Decision Tree model**.

6.1: Comparison of Test Statistics

Comparison of the two models using the same test dataset was conducted earlier. The most salient test results are reproduced here for ease of review.

First for review is the confusion matrices and their derived metrics.

```
print("Logistic Regression Confusion Matrix:")
```

```
## [1] "Logistic Regression Confusion Matrix:"
```

```
print(logit_cm$table)
```

```
##           Reference
## Prediction Died Survived
##   Died      216       41
##   Survived   45       92
```

```
print("Decision Tree Confusion Matrix:")
```

```
## [1] "Decision Tree Confusion Matrix:"
```

```
print(tree_cm$table)
```

```
##           Reference
## Prediction Died Survived
##   Died      222       42
##   Survived   39       91
```

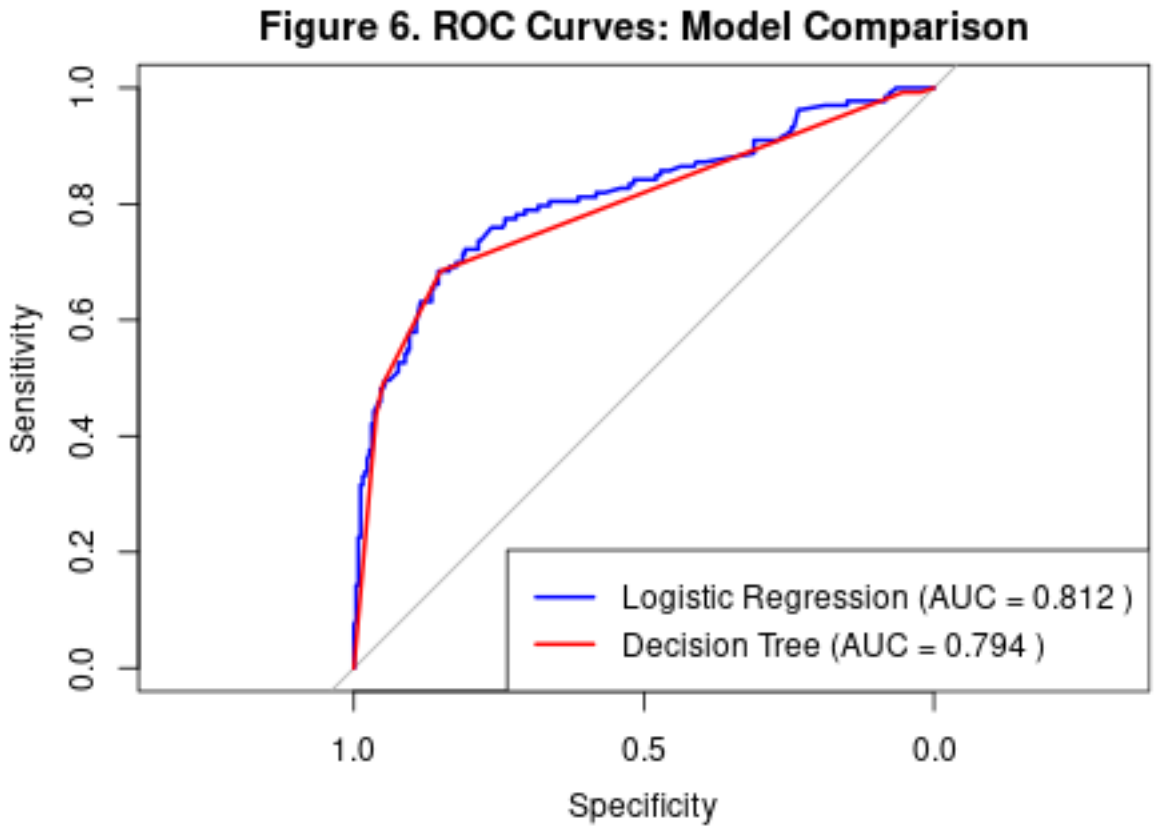
```
metrics_comparison %>%
  knitr::kable(digits = 4,
               caption = "Table 3: Test Set Performance Comparison")
```

Table 7: Table 3: Test Set Performance Comparison

Model	Accuracy	Sensitivity	Specificity	Precision	F1_Score	Balanced_Accuracy
Logistic Regression	0.7817	0.6917	0.8276	0.6715	0.6815	0.7597
Decision Tree	0.7944	0.6842	0.8506	0.7000	0.6920	0.7674

Second is the Area Under the Curve (AUC) plot.

```
plot(logit_roc, col = "blue", lwd = 2, main = "Figure 6. ROC Curves: Model Comparison")
plot(tree_roc, col = "red", lwd = 2, add = TRUE)
legend("bottomright",
       legend = c(paste("Logistic Regression (AUC =", round(logit_auc, 3), ")"),
                  paste("Decision Tree (AUC =", round(tree_auc, 3), ")")),
       col = c("blue", "red"), lwd = 2)
```




```
# AUC comparison table
auc_comparison <- data.frame(
  Model = c("Logistic Regression", "Decision Tree"),
  AUC = c(logit_auc, tree_auc)
)

auc_comparison %>%
  knitr::kable(digits = 4,
               caption = "Table 4: AUC Comparison")
```

Table 8: Table 4: AUC Comparison

Model	AUC
Logistic Regression	0.8123
Decision Tree	0.7935

These test figures illustrate that while performance between the models is similar, the Logistic Regression model has a slight edge in key regards.

1. The Logistic Regression had slightly higher AUC, which is similar to R-Squared in that it indicates how much of the variation is explained by each model.
2. The Logistic Regression had higher Sensitivity (Recall), which gives the designated "positive" outcome–"Survived" in this case–more weight. This is because Recall is composed of all True Positives divided by the sum of all True Positives and all False Negatives (i.e. positives incorrectly identified as negatives). In other words, Recall measures the percentage of all positive outcomes were identified as such. In the context of this dataset, Recall should be given greater consideration, because it measures models by how well they identify survivors of the sinking of the Titanic. Failing to rescue survivors would have the result of them no longer being such, as they are left in the freezing waters of the Atlantic Ocean without rescue.

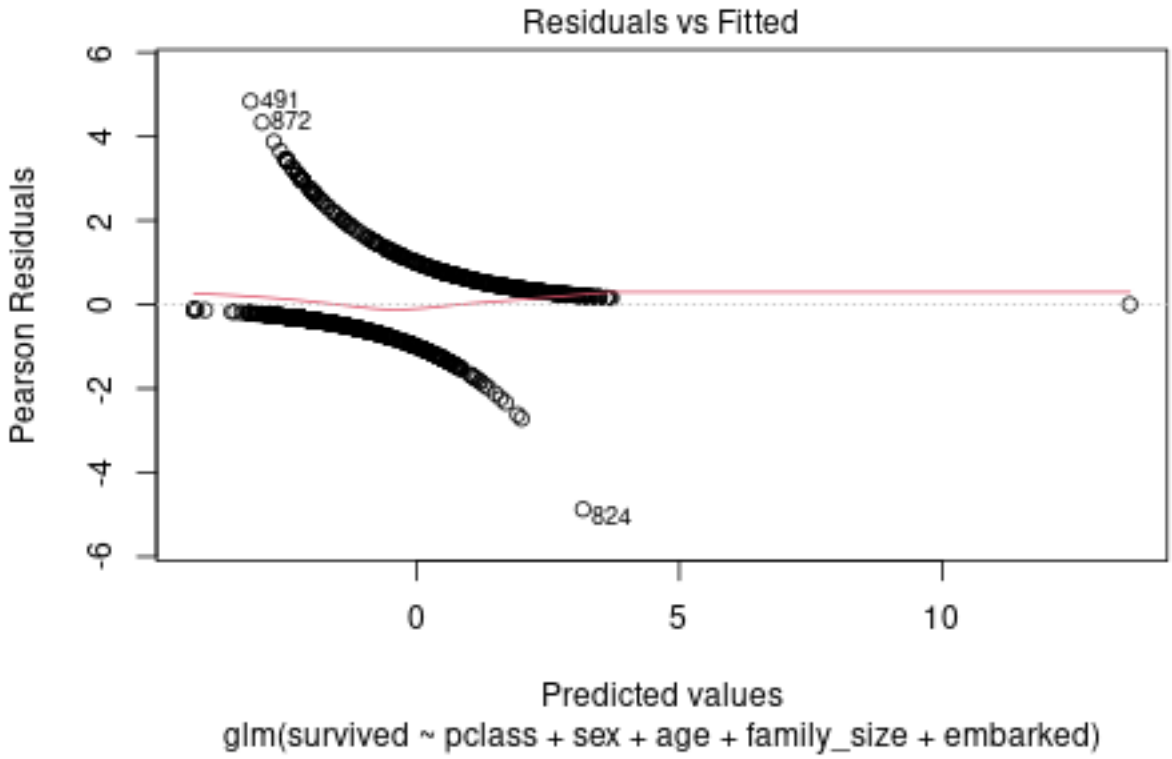
6.2: Analysis of Champion Model Residuals

Because Logistic Models essentially take a regression model's return of a continuous value and transform it to a categorical outcome, such models require particular handling when evaluating their errors.

6.2.1: Champion Model Confusion Matrix Because the models' task is classification, predicting a binary categorical outcome, the most useful measure of "residuals" is a Confusion Matrix. Given the that the two models have comparable Confusion Matrices, the number of false positives and false negatives in the Logistic Regression model seem reasonable.

6.2.2: Champion Model Residuals Below is the plot of the model's residuals.

```
plot(logit_step, which = 1)
```



In Logistic Regression models, there is nonconstant error variance because the "error" is the actual outcome value (in this case, 1 for "Survived" or 0 for "Died"), minus the predicted probability value (between 0 and 1, and seldom exactly 0 or 1). The relationship is expressed mathematically like so.

$$\epsilon_i = Y_i - \hat{\pi}_i$$

On account of this relationship, errors will not be normally distributed, and so this expectation is inapplicable. Classification of observations into one of two categories the goal. It is the application of a cutoff that transforms continuous values between 0 and 1 (inclusive) which yields the categorical outcome, and so it does not matter how far off a given raw model output value (pre-cutoff) is from 0 or 1 as long as it is on the correct side of the cutoff to match the actual value.

6.3: Champion Model Pseudo-R-Squared and Error Review

Because the champion model is a Logistic Regression and its challenger is a Decision Tree, only one of these models even generates errors which are appropriate for measurement with Sum of Squares of Error (SSE) or Residual Mean Squared Error (RMSE).

However, a Pseudo-R-Squared value can be obtained using AUC, as noted above. With an **AUC value of 0.8123, the Logistic Regression has sound performance**, as it can essentially account for 81% of variation using its predictors.

6.4: Champion Model Relative Fit

The Logistic Regression model has a Sensitivity (another term for "Recall") of 0.692. This means that of all actual survivors, the model correctly identifies 69.2% of them. As this value is substantially better than a coin toss, this makes for an effective model.

6.5: Assessment of Logistic Model Assumptions

The Logistic Model does not violate most applicable assumptions.

- The model does not suffer from Multicollinearity. This was confirmed by checking Variable Inflation Factor (VIF) during model construction: it did not yield any concerningly high values (i.e. greater than 5).
- The model has sufficient data, with 1310 observations.
- The model passes linearity concerns. Most of its predictors are continuous, and its output is also continuous.

However, **it does violate the assumption of observation independence**, to a limited degree. The `family_size` predictor variable was synthesized during data cleaning, and combines...

- `sibsp`, the number of siblings or spouses aboard; and...
- `parch`, the number of parents or children aboard.

If a passenger captured in the dataset has a family member (spouse, child, or parent) aboard, and that family member is also captured in the dataset, the `family_size` value of each such passenger will increase by 1.

To understand the degree of this effect, the distribution of `family_size` values should be examined.

```
has_family <- length(which(titanic_train$family_size > 0)) + length(which(titanic_test$family_size > 0))
total_pass <- nrow(titanic_train) + nrow(titanic_test)
pct_w_fam = has_family / total_pass
sprintf('Percent passengers with family aboard = %f', pct_w_fam * 100)
```

```
## [1] "Percent passengers with family aboard = 99.923664"
```

```
print('Training dataset family_size distribution:')
```

```
## [1] "Training dataset family_size distribution:"
```

```
summary(titanic_train$family_size)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   1.000   1.000   1.000   1.886   2.000  11.000       1
```

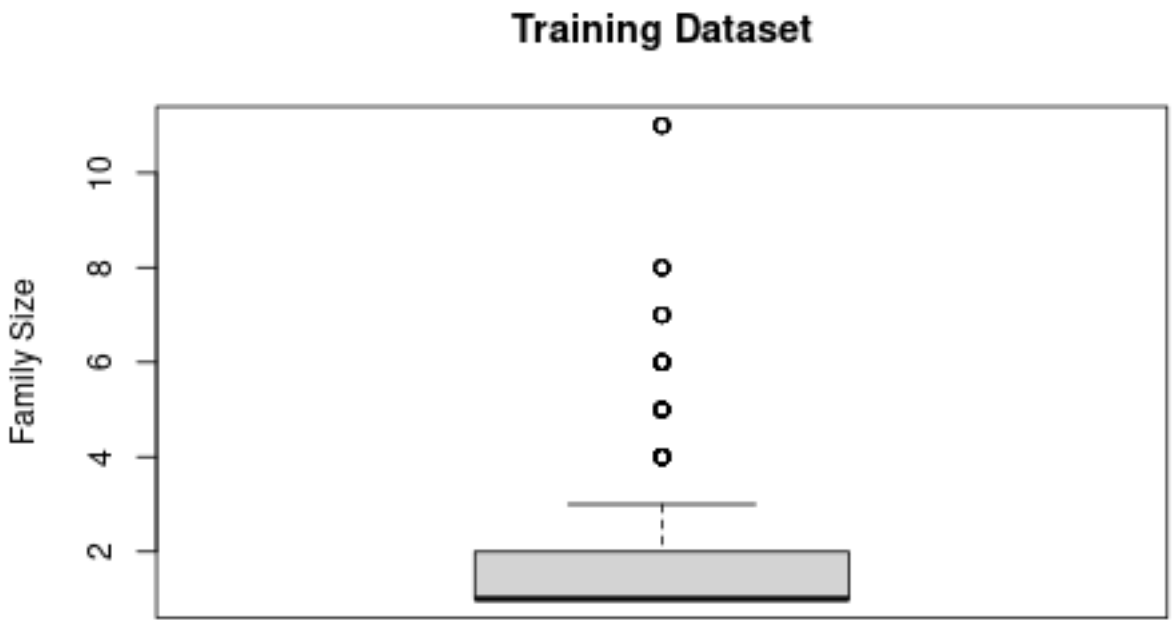
```
print('Testing dataset family_size distribution:')
```

```
## [1] "Testing dataset family_size distribution:"
```

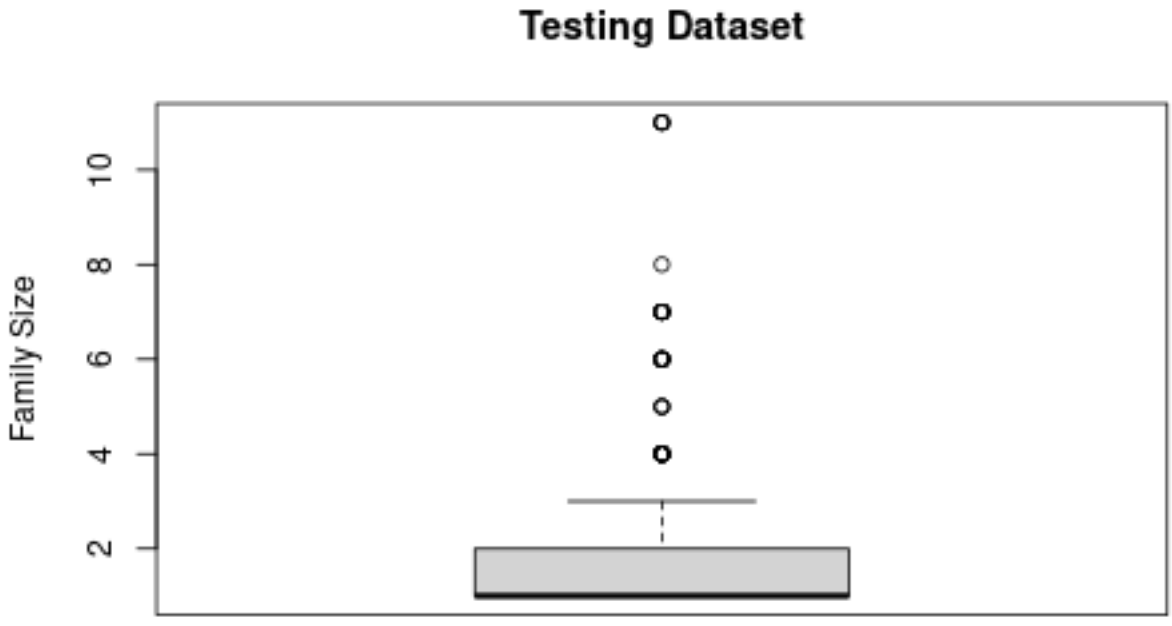
```
summary(titanic_test$family_size)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   1.878   2.000  11.000
```

```
boxplot(titanic_train$family_size, ylab = 'Family Size', main = 'Training Dataset')
```




```
boxplot(titanic_test$family_size, ylab = 'Family Size', main = 'Testing Dataset')
```



Nearly every passenger in the dataset has a family member aboard, meaning there is some degree of observation interdependence for this variable. However, as can be observed in the distribution of values, most passengers had between 1 and 2 family members aboard. Between the relative consistency of the data, and the fact that VIF analysis did not indicate any major issues with this (or any) variable, the violation of this assumption can be tolerated.

6.6: Champion Model Limitations

There are many passengers whose age is unknown. These values had to be imputed: for our analysis, observations with null age values were assigned the median age associated with their sex and class of passage aboard the Titanic. Like all imputations, this is obviously not as if all ages were truly known.

```
age_na <- length(which(is.na(titanic_raw$age)))
sprintf('Number of NULL age values  = %d', age_na)
```

```
## [1] "Number of NULL age values  = 264"
```

```
sprintf('Percent of all values NULL = %f', 100 * (age_na / nrow(titanic_raw)))
```

```
## [1] "Percent of all values NULL = 20.152672"
```

```
# Citations
#   - Gemini to obtain is.na() function (original idea to use to return number of null values in column)
```

As shown above, this imputation affects about 20% of all observations.

Thus, **the Champion model assumes that the imputed values do not vary significantly from the true values, and that any such variation is evenly distributed**. Because at the time, ages in a population were roughly normally distributed, this is a reasonable-enough assumption.

6.7: Champion Model Required Assumptions

Logistic Regression models are very resilient because they do not require many assumptions. Such models only assume that the response variable is categorical and only has 2 states. As the variable to be predicted is whether a Titanic passenger survived or perished after the vessel's sinking, this assumption is met.

The only assumptions required to support this model pertain to the data itself, and would thus affect the Challenger model as well. Namely, those assumptions are...

1. The interrelation between observations as regards `family_size` is consistent enough that it does not seriously undermine use of this variable, and...
2. Actual ages by sex and passage class are evenly distributed enough around median ages by sex and passage class that imputing the latter for null values does not seriously undermine use of this variable.

VII. Ongoing Model Monitoring Plan

- **Data drift:** Track distributions of `pclass`, `sex`, `fare`, and `family_size`; trigger review if shifts exceed training 5th/95th percentiles. If a feature drifts for one period, run a data quality audit and back-test with a challenger model. If drift persists for two consecutive periods, refit the logistic regression on the most recent labeled data while keeping the same feature set; if drift persists after that refit, pause scoring until a new model is validated.

- **Performance:** Recompute accuracy, balanced accuracy, and AUC quarterly. If accuracy < 0.80 or AUC < 0.78 in a period, recalibrate the classification threshold on the latest data and re-score. If the next period remains below these thresholds, retrain on the recent two–three periods. If, after retraining, metrics stay below 0.78 AUC or 0.75 accuracy for two straight checks, retire the champion and fall back to the challenger until a replacement is validated.

- **Stability:** Monitor calibration (Hosmer-Lemeshow) and confusion matrix balance; investigate rising false negatives (missed survivors). If Hosmer-Lemeshow $p < 0.05$ or the false negative rate rises >20% versus baseline, adjust the threshold; if the next review shows the same issue, retrain. If false negatives remain elevated after the retrain, halt deployment until resolved.

- **Process:** Freeze scoring code, log model version/seed, and maintain challenger comparisons on new data. If the champion is paused, route scoring to the challenger or a simple baseline until the fix is validated.

VIII. Conclusion

The stepwise logistic regression is the champion model: it delivers strong discriminatory power, balanced performance, and interpretable odds ratios. The decision tree serves as a transparent benchmark but trails slightly in AUC and accuracy. Monitoring should focus on input drift and sustained predictive performance to ensure continued fitness for purpose.

This project developed and evaluated statistical classification models to predict passenger", "survival on the RMS Titanic. After a comprehensive analysis of data quality, exploratory", "patterns, model diagnostics, and test-set performance, the stepwise logistic regression", "model emerged as the champion model. Its strengths include stable and interpretable", "coefficients, strong discriminatory power (AUC = 0.81), balanced accuracy, and robustness", "against multicollinearity. The model effectively captured key survival determinants such as", "passenger class, sex, age, family structure, and port of embarkation.

The decision tree served as a transparent challenger model, offering rule-based explanations", "that align with well-known Titanic survival dynamics (e.g., higher survival rates among women", "and passengers in first class). Although the tree performed competitively with accuracy and", "AUC close to the logistic regression it exhibited slightly lower generalization performance on", "the hold-out test set, justifying its role as a benchmark rather than the primary model.

Overall, the modeling framework demonstrates that demographic and ticket-related features", "provide meaningful predictive signal for survival classification. The results highlight the", "importance of rigorous feature engineering, proper handling of missing data, and balanced", "evaluation across accuracy, sensitivity, specificity, and AUC. While the champion model is well", "suited for this dataset, limitations such as historical bias, imputation uncertainty, and", "non-linear effects suggest opportunities for future enhancement using ensemble methods or", "calibrated probability models.

The final recommendation is to adopt the logistic regression model as the primary forecasting", "tool, supported by ongoing monitoring of input drift, predictive performance decay, and model", "assumptions. With appropriate governance, this modeling solution is fit for purpose and", "provides a reproducible, transparent, and statistically grounded approach to survival", "prediction on the Titanic dataset.

Bibliography

Kaggle. (n.d.). *Titanic: Machine Learning from Disaster*. Retrieved from https://www.kaggle.com/competitions/titanic. Used as the primary source of the Titanic survival dataset and baseline feature descriptions.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in R* (2nd ed.). Springer. Key reference for logistic regression, model selection, and performance evaluation concepts applied in this report.

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer. Used for guidance on train/test splitting, confusion matrices, ROC curves, AUC, and model comparison in classification problems.

Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied Logistic Regression* (3rd ed.). Wiley. Reference for interpreting odds ratios, assessing goodness-of-fit, and applying the Hosmer–Lemeshow test for the champion logistic model.

Harrell, F. E. (2015). *Regression Modeling Strategies* (2nd ed.). Springer. Conceptual support for model building, handling of nonlinearity, and the use of diagnostic plots for residuals and influential observations.

Kuhn, M. (2008). *caret: Classification and Regression Training* [R package]. Comprehensive R framework used for confusion matrices and model performance metrics; see CRAN documentation for implementation details.

Robin, X., et al. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12(77). Used for ROC curve plotting and AUC computation for the logistic regression and decision tree models.

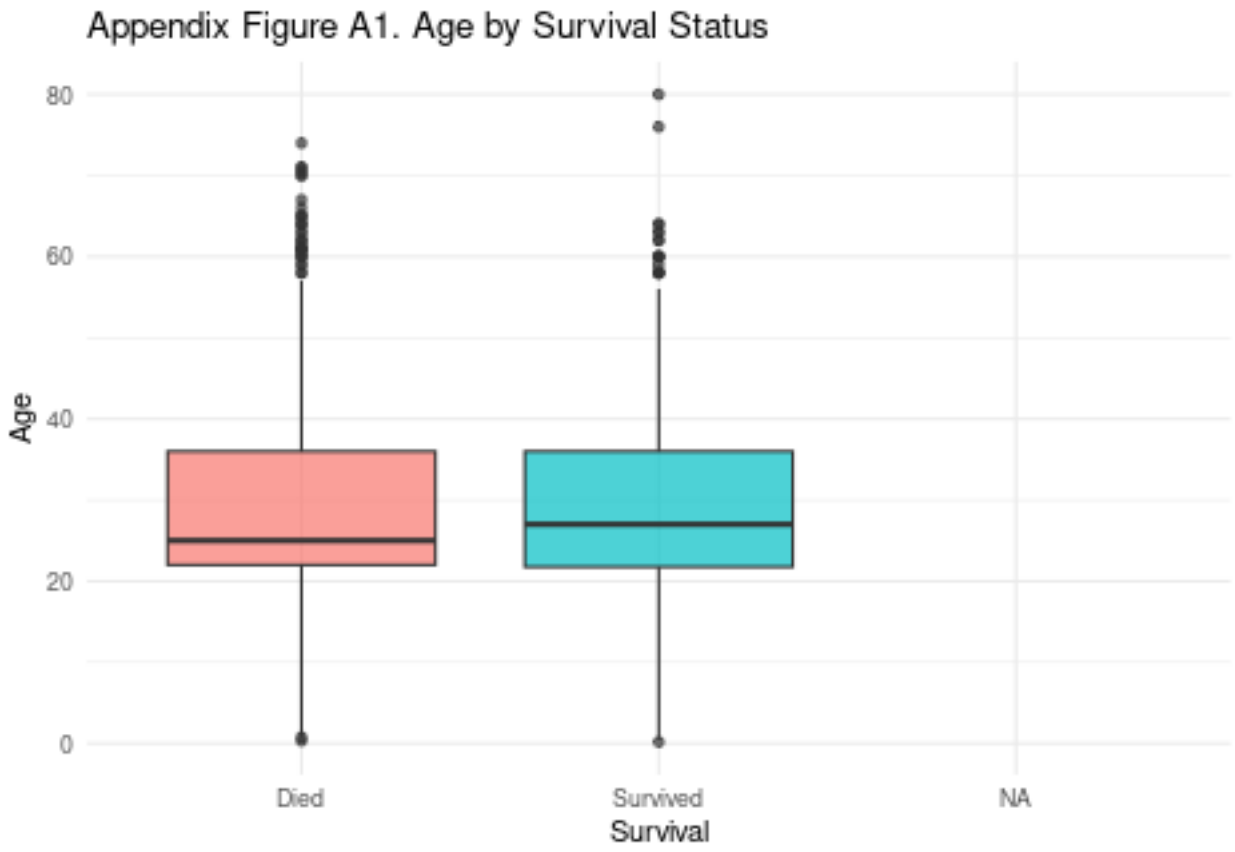
Therneau, T., & Atkinson, B. (2024). *rpart: Recursive Partitioning and Regression Trees* [R package]. Implemented the decision tree challenger model for survival classification and rule-based interpretation.

Milborrow, S. (2024). *rpart.plot: Plot 'rpart' Models* [R package]. Used to visualize the decision tree structure and communicate simple, interpretable survivor/non-survivor rules.

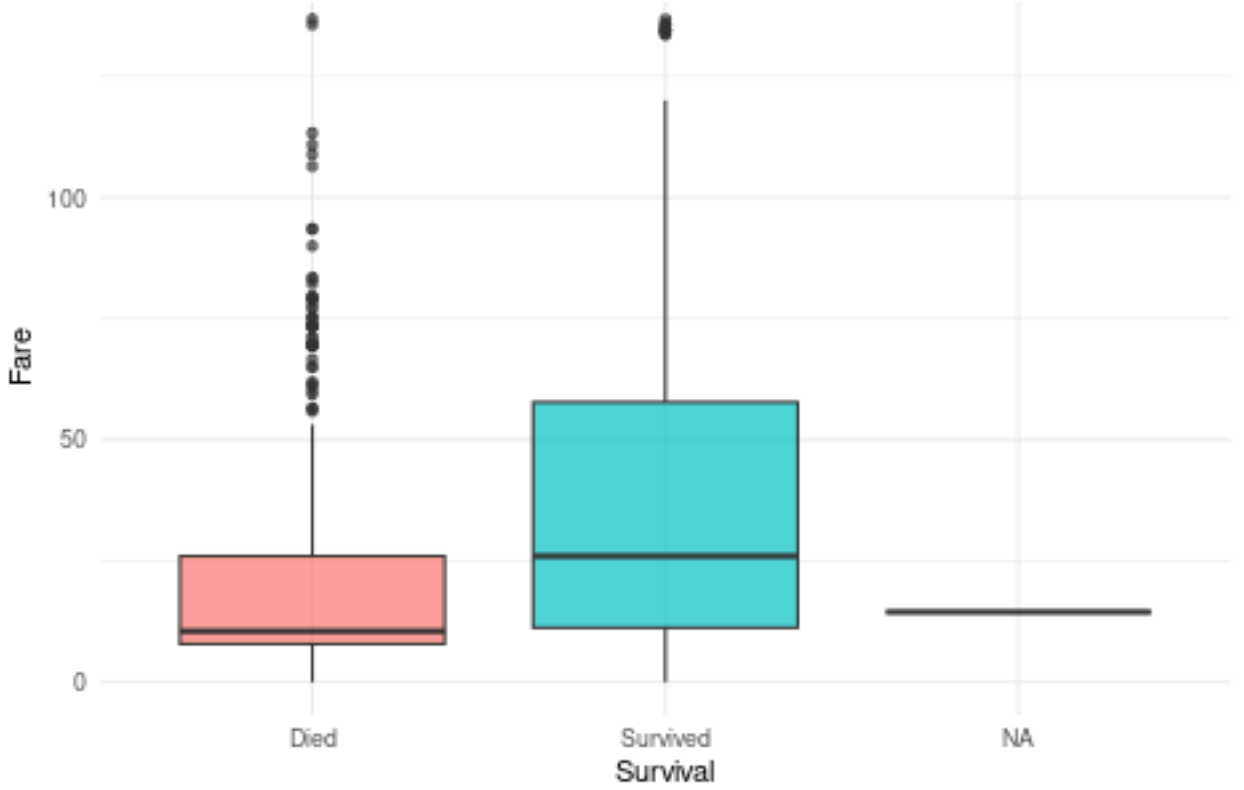
R Core Team. (2024). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. The statistical computing environment used for all data manipulation, modeling, and graphics in this project.

Appendix

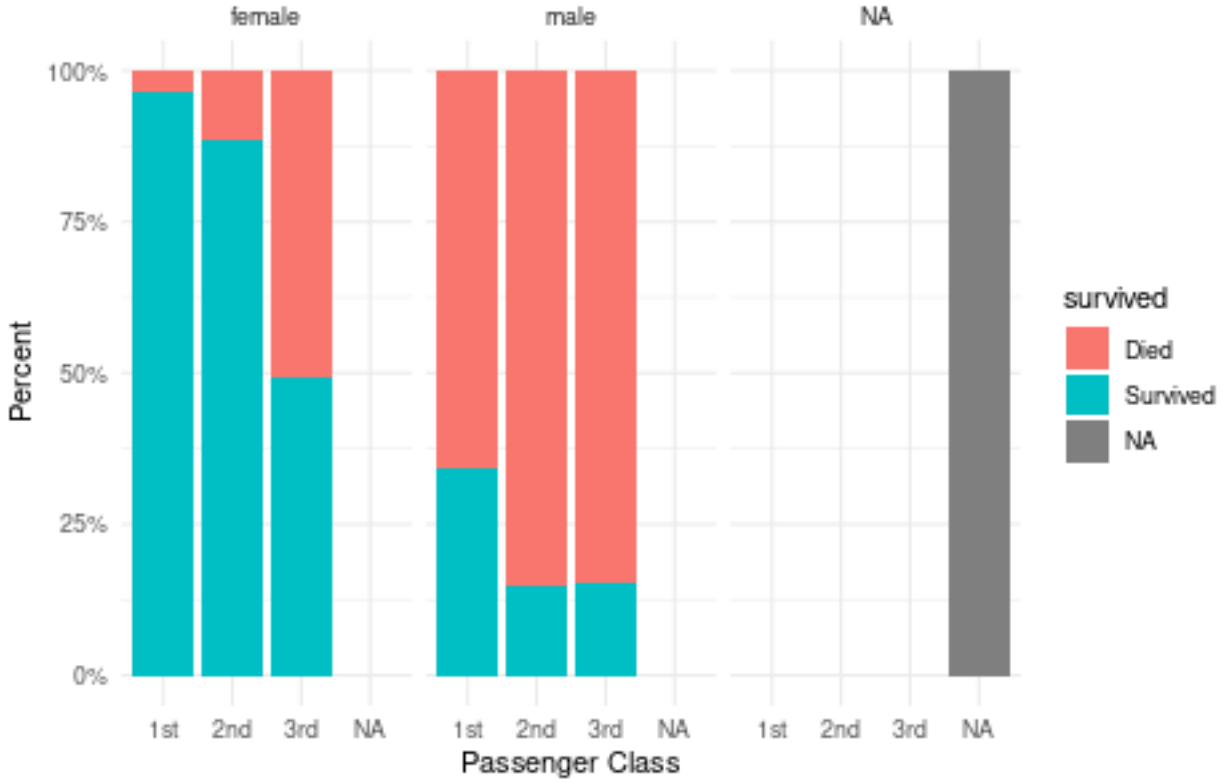
A1. Additional Exploratory Data Analysis

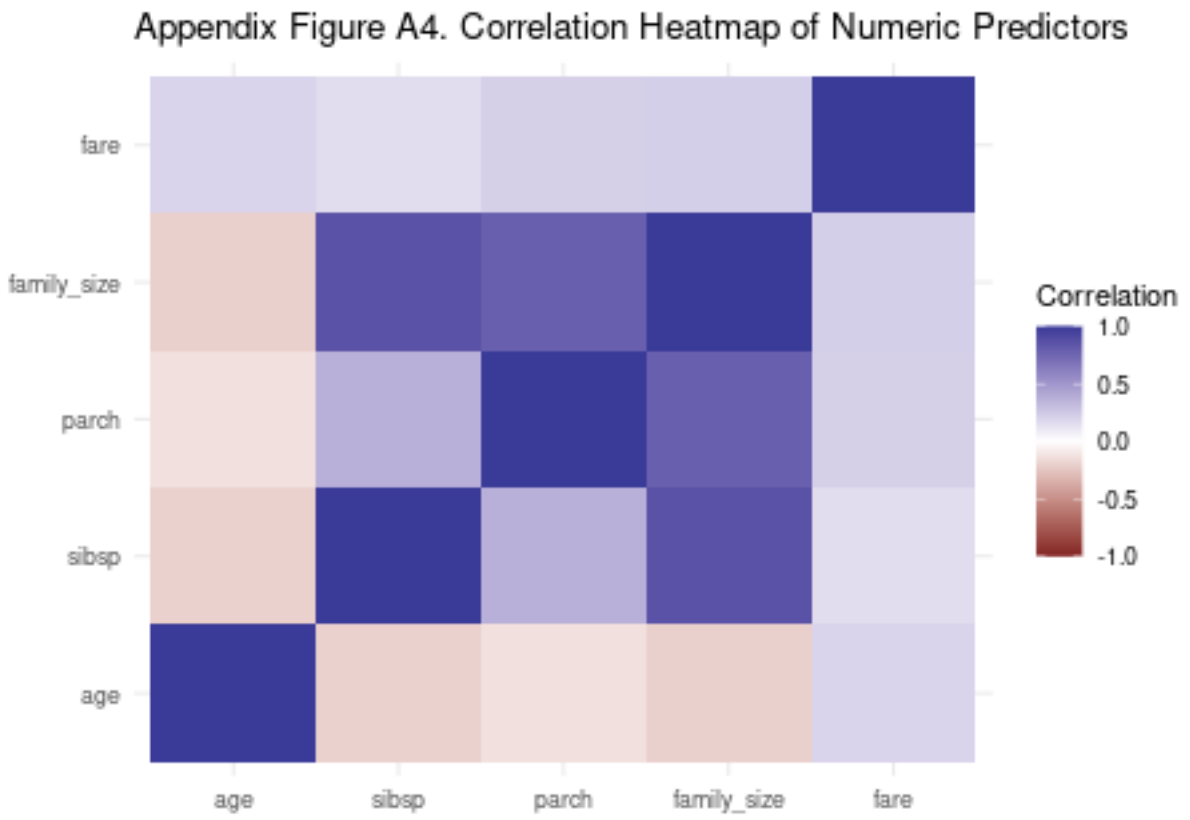


Appendix Figure A2. Fare by Survival Status (Truncated at 95th Percentile)



Appendix Figure A3. Survival Share by Class and Sex





Interpretation (EDA): The additional plots confirm strong differences in age and fare distributions across survival groups and highlight interaction patterns between class and sex. The correlation heatmap shows only moderate correlations among numeric predictors, which is consistent with the low VIF values reported in the main text.

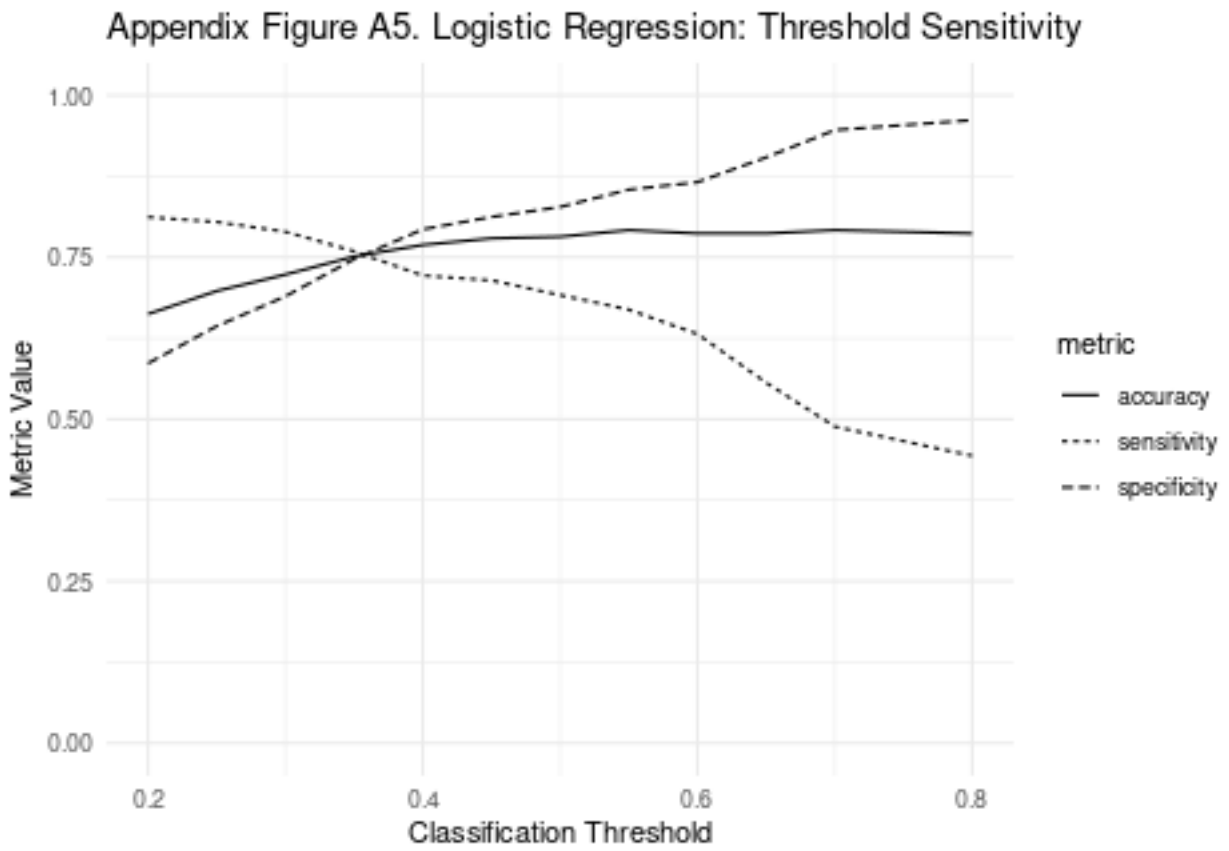
A2. Detailed Confusion Matrices and Threshold Sensitivity

Table 9: Appendix Table A1. Confusion Matrix – Logistic Regression (Cutoff = 0.50)

	Died	Survived
Died	216	41
Survived	45	92

Table 10: Appendix Table A2. Confusion Matrix – Decision Tree

	Died	Survived
Died	222	42
Survived	39	91



Interpretation (Thresholds): Varying the cutoff between 0.20 and 0.80 shows the usual trade-off between sensitivity and specificity. The default 0.50 threshold achieves a good balance, but alternative cutoffs could be chosen if business priorities required fewer false negatives or fewer false positives.

A3. Additional ROC / AUC Details

Table 11: Appendix Table A3. Area Under the ROC Curve (AUC) by Model

Model	AUC
Logistic Regression	0.812
Decision Tree	0.794

Interpretation (AUC): Both models substantially outperform random classification, with AUC values around 0.80. The logistic regression shows a slightly higher AUC than the decision tree, which supports its selection as the champion model in the main text.

A4. Reproducibility Notes

Reproducibility: All results in the report can be regenerated by running this R Markdown document from top to bottom. Key modeling choices include the 70/30 train/test split with set.seed(1023), median imputations for age and fare, and the predictor set used in the logistic regression and decision tree. This appendix collects supporting plots and tables that were omitted from the main body for brevity but may be helpful for technical reviewers.