

```

# Load Required Libraries
library(testthat)
library(tidyverse)
library(mgcv) # For GAM models
library(ggplot2)

# Code your answers as TRUE or FALSE.

prob.1.1 = FALSE
prob.1.2 = TRUE
prob.1.3 = TRUE
prob.1.4 = FALSE
prob.1.5 = TRUE

# your code here

# Test Cell
# Make sure your answers are booleans!
# This cell has hidden test cases that will run after submission.

# Test Cell
# This cell has hidden test cases that will run after submission.

# Test Cell
# This cell has hidden test cases that will run after submission.

# Test Cell
# This cell has hidden test cases that will run after submission.

# Test Cell
# This cell has hidden test cases that will run after submission.

# Load in the Data
titanic = read.csv("titanic.csv", sep=",")
head(titanic)

titanic.train = NA
titanic.test = NA

# your code here

# Keep only selected columns
titanic = titanic[,c("Survived", "Pclass", "Sex", "Age", "Fare")]

# Remove rows with NA values
titanic = titanic[complete.cases(titanic),]

# Set factors as categorical
titanic$Survived = factor(titanic$Survived)
titanic$Pclass = factor(titanic$Pclass)

# Create test/train split
index = 1:nrow(titanic)
test_index = seq(5,nrow(titanic),5)

```

```

train_index = setdiff(index, test_index)

titanic.train = titanic[train_index,]
titanic.test = titanic[test_index,]

# Test Cell
# This cell has hidden test cases that will run after submission.
if(!test_that("Checking DataFrame Size", {expect_equal(nrow(titanic), 714)
                                                    expect_equal(nrow(titanic.train), 572)}))
{
  print("Incorrect Dataset sizes. Make sure these are correct, or else your
modelling could be incorrect.")
}

titanic.gam = NA
insig.predictors = c()

# your code here
# Fit model
# Fit GAM model
titanic.gam <- mgcv::gam(formula = Survived ~ Pclass + Sex +
                        s(Age) + s(Fare),
                        data = titanic.train,
                        family = "binomial",
                        method = "REML")

# Check model summary
summary(titanic.gam)

# Identify insignificant predictors
insig.predictors <- c()
if(summary(titanic.gam)$s.table[, "edf"] <= 1){
  insig.predictors <- c(insig.predictors,
                        rownames(summary(titanic.gam)$s.table[summary(titanic.gam)
$s.table["edf"] <= 1,]))
}

# Another option:
insig_index <- which(summary(titanic.gam)$s.table[, 4] > 0.05)
insig.predictors <- rownames(summary(titanic.gam)$s.table)[insig_index]

# Test Cell
# This cell has hidden test cases that will run after submission.

# Test Cell
# This cell has hidden test cases that will run after submission.

age.is.linear = NA
fare.is.linear = NA

# your code here

# Plot smooth terms
plot(titanic.gam)

# Check edf
summary(titanic.gam)

```

```

# Assess linearity
age.is.linear <- TRUE
fare.is.linear <- TRUE

# Test Cell
# This cell has hidden test cases that will run after submission.

# Test Cell
# This cell has hidden test cases that will run after submission.

gam.acc = NA
gam.prec = NA
gam.rec = NA
gam.f1 = NA

# your code here
# Make predictions on test set
test_pred <- predict(titanic.gam, newdata = titanic.test, type = "response")

# Convert predictions to 0/1 based on 0.5 threshold
test_pred <- ifelse(test_pred > 0.5, 1, 0)

# Calculate accuracy
gam.acc <- mean(test_pred == titanic.test$Survived)

# Calculate precision
tp <- sum((test_pred == 1) & (titanic.test$Survived == 1))
fp <- sum((test_pred == 1) & (titanic.test$Survived == 0))
gam.prec <- tp / (tp + fp)

# Calculate recall
tp <- sum((test_pred == 1) & (titanic.test$Survived == 1))
fn <- sum((test_pred == 0) & (titanic.test$Survived == 1))
gam.rec <- tp / (tp + fn)

# Calculate F1 score
gam.f1 <- 2 * (gam.prec * gam.rec) / (gam.prec + gam.rec)

# Test Cell
# This cell has hidden test cases that will run after submission.

```