group project titanic

leo

library(here)

Warning: package 'here' was built under R version 4.3.3

here() starts at C:/Users/Leonel/Desktop/MSDA/MSDA/MS Data Analytics/Current Class/DA 6213/final project

library(survival)  
library(survminer)

Warning: package 'survminer' was built under R version 4.3.3

Loading required package: ggplot2

Warning: package 'ggplot2' was built under R version 4.3.3

Loading required package: ggpubr

Warning: package 'ggpubr' was built under R version 4.3.3

Attaching package: 'survminer'

The following object is masked from 'package:survival':  
  
 myeloma

library(dplyr)

Warning: package 'dplyr' was built under R version 4.3.3

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':  
  
 filter, lag

The following objects are masked from 'package:base':  
  
 intersect, setdiff, setequal, union

library(caret)

Warning: package 'caret' was built under R version 4.3.3

Loading required package: lattice

Warning: package 'lattice' was built under R version 4.3.3

Attaching package: 'caret'

The following object is masked from 'package:survival':  
  
 cluster

library(randomForest)

Warning: package 'randomForest' was built under R version 4.3.3

randomForest 4.7-1.1

Type rfNews() to see new features/changes/bug fixes.

Attaching package: 'randomForest'

The following object is masked from 'package:dplyr':  
  
 combine

The following object is masked from 'package:ggplot2':  
  
 margin

library(pROC)

Warning: package 'pROC' was built under R version 4.3.3

Type 'citation("pROC")' for a citation.

Attaching package: 'pROC'

The following objects are masked from 'package:stats':  
  
 cov, smooth, var

library(survival)   
library(survminer)

library(dplyr)  
  
titanic\_df <- read.csv('Titanic-Dataset.csv')  
  
titanic\_df$Sex <- as.factor(titanic\_df$Sex)  
titanic\_df$Pclass <- as.factor(titanic\_df$Pclass)  
titanic\_df$Embarked <- as.factor(titanic\_df$Embarked)  
titanic\_df$Survived <- as.factor(titanic\_df$Survived)  
titanic\_df$CabinDeck <- factor(ifelse(nchar(as.character(titanic\_df$Cabin)) > 0,   
 substr(as.character(titanic\_df$Cabin), 1, 1),   
 'Unknown'))  
titanic\_df$FamilySize <- titanic\_df$SibSp + titanic\_df$Parch + 1

# Load the Titanic dataset   
titanic\_df <- read.csv("Titanic-Dataset.csv", stringsAsFactors = FALSE)   
   
# Prepare variables: create FarePerAge, CabinLocation, FamilySize, and IsAlone   
titanic\_df$FarePerAge <- with(titanic\_df, ifelse(!is.na(Age) & Age > 0, Fare / Age, NA))   
titanic\_df$CabinLocation <- ifelse(!is.na(titanic\_df$Cabin) & titanic\_df$Cabin != "",   
 substr(titanic\_df$Cabin, 1, 1),   
 "Unknown")   
titanic\_df$FamilySize <- with(titanic\_df, SibSp + Parch + 1)   
titanic\_df$IsAlone <- ifelse(titanic\_df$FamilySize == 1, 1, 0)   
   
# Convert variables to factors where needed   
titanic\_df$Embarked <- as.factor(titanic\_df$Embarked)   
titanic\_df$CabinLocation <- as.factor(titanic\_df$CabinLocation)   
titanic\_df$IsAlone <- as.factor(titanic\_df$IsAlone)   
   
# Run the logistic regression analysis with fixed CabinLocation   
logit\_model <- glm(Survived ~ Embarked + FarePerAge + CabinLocation + FamilySize + IsAlone,   
 data = titanic\_df, family = binomial)   
   
# Output the summary of the logistic regression model   
print(summary(logit\_model))

Call:  
glm(formula = Survived ~ Embarked + FarePerAge + CabinLocation +   
 FamilySize + IsAlone, family = binomial, data = titanic\_df)  
  
Coefficients:  
 Estimate Std. Error z value Pr(>|z|)   
(Intercept) 15.33188 624.16547 0.025 0.98040   
EmbarkedC -13.66272 624.16509 -0.022 0.98254   
EmbarkedQ -14.12731 624.16523 -0.023 0.98194   
EmbarkedS -14.17328 624.16508 -0.023 0.98188   
FarePerAge 0.05699 0.02162 2.637 0.00838 \*\*   
CabinLocationB 0.55126 0.70728 0.779 0.43574   
CabinLocationC -0.26354 0.68228 -0.386 0.69930   
CabinLocationD 0.58131 0.74314 0.782 0.43408   
CabinLocationE 0.80566 0.74633 1.079 0.28037   
CabinLocationF 0.32430 0.90304 0.359 0.71951   
CabinLocationG -0.53735 1.19115 -0.451 0.65191   
CabinLocationT -14.35597 882.74359 -0.016 0.98702   
CabinLocationUnknown -0.90934 0.62340 -1.459 0.14466   
FamilySize -0.27987 0.09570 -2.925 0.00345 \*\*   
IsAlone1 -1.13378 0.25417 -4.461 8.17e-06 \*\*\*  
---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
  
(Dispersion parameter for binomial family taken to be 1)  
  
 Null deviance: 964.52 on 713 degrees of freedom  
Residual deviance: 839.03 on 699 degrees of freedom  
 (177 observations deleted due to missingness)  
AIC: 869.03  
  
Number of Fisher Scoring iterations: 13

# Calculate Odds Ratios and 95% Confidence Intervals  
or\_values <- exp(coef(logit\_model))  
ci\_values <- exp(confint(logit\_model))

Waiting for profiling to be done...

Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
  
Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

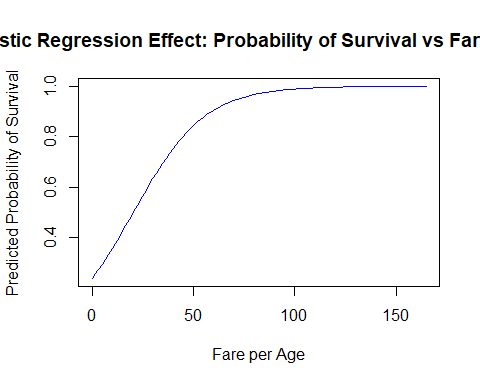
print(or\_values)

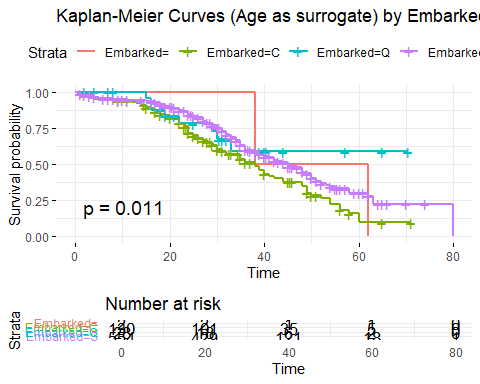
(Intercept) EmbarkedC EmbarkedQ   
 4.555634e+06 1.165077e-06 7.321316e-07   
 EmbarkedS FarePerAge CabinLocationB   
 6.992357e-07 1.058648e+00 1.735439e+00   
 CabinLocationC CabinLocationD CabinLocationE   
 7.683253e-01 1.788387e+00 2.238181e+00   
 CabinLocationF CabinLocationG CabinLocationT   
 1.383064e+00 5.842969e-01 5.824782e-07   
CabinLocationUnknown FamilySize IsAlone1   
 4.027887e-01 7.558803e-01 3.218148e-01

print(ci\_values)

2.5 % 97.5 %  
(Intercept) 2.693116e-36 NA  
EmbarkedC NA 3.449370e+36  
EmbarkedQ NA 1.784065e+36  
EmbarkedS NA 2.105507e+36  
FarePerAge 1.018352e+00 1.108769e+00  
CabinLocationB 4.179260e-01 6.975283e+00  
CabinLocationC 1.923169e-01 2.912960e+00  
CabinLocationD 4.054581e-01 7.778211e+00  
CabinLocationE 5.049584e-01 9.804868e+00  
CabinLocationF 2.330402e-01 8.468545e+00  
CabinLocationG 5.063128e-02 6.648300e+00  
CabinLocationT NA 6.110133e+71  
CabinLocationUnknown 1.120350e-01 1.363418e+00  
FamilySize 6.209936e-01 9.052443e-01  
IsAlone1 1.946926e-01 5.280384e-01

# Create a logistic regression effect plot for FarePerAge  
# We predict probability across a range of FarePerAge values while holding other variables constant (using median or reference levels)  
fare\_range <- seq(min(titanic\_df$FarePerAge, na.rm = TRUE), max(titanic\_df$FarePerAge, na.rm = TRUE), length.out = 100)  
# use typical values: Embarked = most common, CabinLocation = most common, FamilySize = median, IsAlone = most common  
common\_embarked <- levels(titanic\_df$Embarked)[which.max(table(titanic\_df$Embarked))]  
common\_cabin <- names(which.max(table(titanic\_df$CabinLocation)))  
median\_family <- median(titanic\_df$FamilySize, na.rm = TRUE)  
common\_alone <- names(which.max(table(titanic\_df$IsAlone)))  
  
test\_data <- data.frame(FarePerAge = fare\_range,  
 Embarked = factor(common\_embarked, levels=levels(titanic\_df$Embarked)),  
 CabinLocation = factor(common\_cabin, levels=levels(titanic\_df$CabinLocation)),  
 FamilySize = median\_family,  
 IsAlone = factor(common\_alone, levels=levels(titanic\_df$IsAlone)))  
  
test\_data$predicted\_prob <- predict(logit\_model, newdata = test\_data, type = 'response')  
  
# Plot the logistic regression effect for FarePerAge  
plot(test\_data$FarePerAge, test\_data$predicted\_prob, type='l', col='blue',  
 main='Logistic Regression Effect: Probability of Survival vs FarePerAge',  
 xlab='Fare per Age', ylab='Predicted Probability of Survival')  
  
  
# Kaplan-Meier Curves  
# Note: The Titanic dataset does not have an explicit time-to-event variable.   
# We use Age as a surrogate for time for demonstration purposes.  
# Create a Surv object with Age as time and Survived as event indicator.  
# Warning: This use is not a true survival analysis based on follow-up time.  
km\_fit\_embarked <- survfit(Surv(Age, Survived) ~ Embarked, data=titanic\_df)  
  
# Plot Kaplan-Meier curves by Embarked category  
km\_plot <- ggsurvplot(km\_fit\_embarked, data=titanic\_df, risk.table = TRUE,   
 pval = TRUE, ggtheme = theme\_minimal(),  
 title = 'Kaplan-Meier Curves (Age as surrogate) by Embarked')  
print(km\_plot)





# Detailed Survival Rates by each Category:  
# For Embarked:  
survival\_rates\_embarked <- aggregate(Survived ~ Embarked, data=titanic\_df, FUN=function(x){sum(x, na.rm = TRUE)/length(x)})  
print(survival\_rates\_embarked)

Embarked Survived  
1 1.0000000  
2 C 0.5535714  
3 Q 0.3896104  
4 S 0.3369565

# For Cabin Location:  
survival\_rates\_cabin <- aggregate(Survived ~ CabinLocation, data=titanic\_df, FUN=function(x){sum(x, na.rm = TRUE)/length(x)})  
print(survival\_rates\_cabin)

CabinLocation Survived  
1 A 0.4666667  
2 B 0.7446809  
3 C 0.5932203  
4 D 0.7575758  
5 E 0.7500000  
6 F 0.6153846  
7 G 0.5000000  
8 T 0.0000000  
9 Unknown 0.2998544

# For Family Size:  
survival\_rates\_family <- aggregate(Survived ~ FamilySize, data=titanic\_df, FUN=function(x){sum(x, na.rm = TRUE)/length(x)})  
print(survival\_rates\_family)

FamilySize Survived  
1 1 0.3035382  
2 2 0.5527950  
3 3 0.5784314  
4 4 0.7241379  
5 5 0.2000000  
6 6 0.1363636  
7 7 0.3333333  
8 8 0.0000000  
9 11 0.0000000

# For IsAlone:  
survival\_rates\_alone <- aggregate(Survived ~ IsAlone, data=titanic\_df, FUN=function(x){sum(x, na.rm = TRUE)/length(x)})  
print(survival\_rates\_alone)

IsAlone Survived  
1 0 0.5056497  
2 1 0.3035382

# Summary statistics for predictors and outcome  
summary\_stats <- summary(titanic\_df[, c('Age', 'Fare', 'FarePerAge', 'FamilySize')])  
print(summary\_stats)

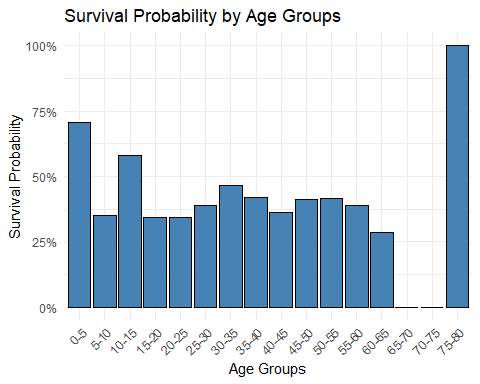
Age Fare FarePerAge FamilySize   
 Min. : 0.42 Min. : 0.00 Min. : 0.0000 Min. : 1.000   
 1st Qu.:20.12 1st Qu.: 7.91 1st Qu.: 0.3424 1st Qu.: 1.000   
 Median :28.00 Median : 14.45 Median : 0.5652 Median : 1.000   
 Mean :29.70 Mean : 32.20 Mean : 2.3918 Mean : 1.905   
 3rd Qu.:38.00 3rd Qu.: 31.00 3rd Qu.: 1.6739 3rd Qu.: 2.000   
 Max. :80.00 Max. :512.33 Max. :164.7283 Max. :11.000   
 NA's :177 NA's :177

# Completed advanced analyses.

titanic\_df <- read.csv('Titanic-Dataset.csv')  
titanic\_df$Sex <- as.factor(titanic\_df$Sex)  
titanic\_df$Pclass <- as.factor(titanic\_df$Pclass)  
titanic\_df$Embarked <- as.factor(titanic\_df$Embarked)  
titanic\_df$Survived <- as.factor(titanic\_df$Survived)  
titanic\_df$CabinDeck <- factor(ifelse(nchar(as.character(titanic\_df$Cabin)) > 0,   
 substr(as.character(titanic\_df$Cabin), 1, 1),   
 'Unknown'))  
titanic\_df$FamilySize <- titanic\_df$SibSp + titanic\_df$Parch + 1  
titanic\_df$IsAlone <- factor(ifelse(titanic\_df$FamilySize == 1, 1, 0))  
  
survivors <- titanic\_df %>%  
 filter(Survived == 1) %>%  
 select(PassengerId, Name, Sex, Pclass, Embarked, CabinDeck, FamilySize, IsAlone, Age, Fare) %>%  
 na.omit()  
  
survivor\_groups <- survivors %>%  
 group\_by(Sex, Pclass, Embarked, CabinDeck, FamilySize, IsAlone) %>%  
 summarise(  
 Count = n(),  
 Age = median(Age),  
 Fare = median(Fare),  
 PassengerIds = paste(sort(PassengerId), collapse = ", "),  
 .groups = "drop"  
 )  
  
model\_data <- titanic\_df %>%  
 select(Survived, Sex, Pclass, Age, Fare, Embarked, CabinDeck, FamilySize, IsAlone) %>%  
 na.omit()  
  
set.seed(123)  
rf\_model <- randomForest(Survived ~ ., data = model\_data, ntree = 1000)  
  
survivor\_groups$survival\_prob <- predict(rf\_model, survivor\_groups, type = "prob")[,2]  
  
best\_combinations <- survivor\_groups %>%  
 arrange(desc(survival\_prob)) %>%  
 mutate(survival\_prob = round(survival\_prob, 3))  
  
print(best\_combinations)

# A tibble: 102 × 11  
 Sex Pclass Embarked CabinDeck FamilySize IsAlone Count Age Fare  
 <fct> <fct> <fct> <fct> <dbl> <fct> <int> <dbl> <dbl>  
 1 female 1 S E 2 0 3 33 55   
 2 female 1 C B 2 0 4 31.5 74.5  
 3 female 1 C E 3 0 2 39.5 109.   
 4 female 1 C D 2 0 6 50.5 78.3  
 5 female 1 C E 1 1 2 35.5 95.7  
 6 female 1 S B 1 1 6 29.5 86.5  
 7 female 1 S B 2 0 3 17 211.   
 8 female 1 C B 3 0 1 22 49.5  
 9 female 1 S D 2 0 3 51 78.0  
10 female 1 S C 2 0 5 35 83.5  
# ℹ 92 more rows  
# ℹ 2 more variables: PassengerIds <chr>, survival\_prob <dbl>

library(dplyr)  
library(ggplot2)  
  
titanic\_df$Survived <- as.numeric(as.character(titanic\_df$Survived))  
  
age\_survival <- titanic\_df %>%  
 filter(!is.na(Age), !is.na(Survived)) %>%  
 mutate(age\_group = cut(Age,   
 breaks = seq(0, 80, by = 5),  
 labels = paste(seq(0, 75, by = 5),   
 seq(5, 80, by = 5),  
 sep = "-"))) %>%  
 group\_by(age\_group) %>%  
 summarise(survival\_prob = mean(Survived))  
  
ggplot(age\_survival, aes(x = age\_group, y = survival\_prob)) +  
 geom\_col(fill = "steelblue", color = "black") +  
 labs(x = "Age Groups",  
 y = "Survival Probability",  
 title = "Survival Probability by Age Groups") +  
 theme\_minimal() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))



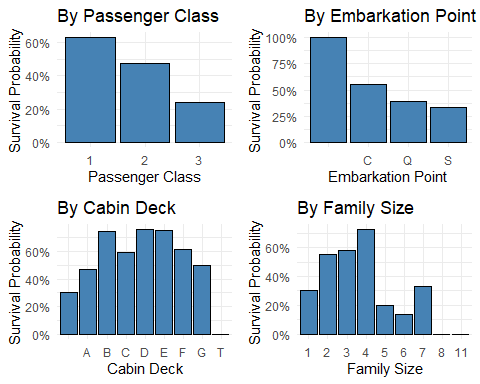
library(dplyr)  
library(ggplot2)  
library(gridExtra)

Attaching package: 'gridExtra'

The following object is masked from 'package:randomForest':  
  
 combine

The following object is masked from 'package:dplyr':  
  
 combine

# Passenger Class  
pclass\_survival <- titanic\_df %>%  
 filter(!is.na(Survived)) %>%  
 mutate(Survived = as.numeric(as.character(Survived))) %>%  
 group\_by(Pclass) %>%  
 summarise(survival\_prob = mean(Survived))  
  
p1 <- ggplot(pclass\_survival, aes(x = factor(Pclass), y = survival\_prob)) +  
 geom\_col(fill = "steelblue", color = "black") +  
 labs(x = "Passenger Class", y = "Survival Probability", title = "By Passenger Class") +  
 theme\_minimal() +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))  
  
# Embarkation Point  
embarked\_survival <- titanic\_df %>%  
 filter(!is.na(Survived), !is.na(Embarked)) %>%  
 mutate(Survived = as.numeric(as.character(Survived))) %>%  
 group\_by(Embarked) %>%  
 summarise(survival\_prob = mean(Survived))  
  
p2 <- ggplot(embarked\_survival, aes(x = Embarked, y = survival\_prob)) +  
 geom\_col(fill = "steelblue", color = "black") +  
 labs(x = "Embarkation Point", y = "Survival Probability", title = "By Embarkation Point") +  
 theme\_minimal() +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))  
  
# Cabin Deck  
titanic\_df$Deck <- substr(titanic\_df$Cabin, 1, 1)  
deck\_survival <- titanic\_df %>%  
 filter(!is.na(Survived), !is.na(Deck)) %>%  
 mutate(Survived = as.numeric(as.character(Survived))) %>%  
 group\_by(Deck) %>%  
 summarise(survival\_prob = mean(Survived))  
  
p3 <- ggplot(deck\_survival, aes(x = Deck, y = survival\_prob)) +  
 geom\_col(fill = "steelblue", color = "black") +  
 labs(x = "Cabin Deck", y = "Survival Probability", title = "By Cabin Deck") +  
 theme\_minimal() +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))  
  
# Family Size  
titanic\_df$FamilySize <- titanic\_df$SibSp + titanic\_df$Parch + 1  
family\_survival <- titanic\_df %>%  
 filter(!is.na(Survived)) %>%  
 mutate(Survived = as.numeric(as.character(Survived))) %>%  
 group\_by(FamilySize) %>%  
 summarise(survival\_prob = mean(Survived))  
  
p4 <- ggplot(family\_survival, aes(x = factor(FamilySize), y = survival\_prob)) +  
 geom\_col(fill = "steelblue", color = "black") +  
 labs(x = "Family Size", y = "Survival Probability", title = "By Family Size") +  
 theme\_minimal() +  
 scale\_y\_continuous(labels = scales::percent\_format(accuracy = 1))  
  
# Arrange all plots in a grid  
grid.arrange(p1, p2, p3, p4, ncol = 2)



**The plots show:**

* By Passenger Class: First class passengers had the highest survival rate, followed by second class, with third class having the lowest survival probability
* By Embarkation Point: Passengers who embarked from ‘C’ (Cherbourg) had the highest survival rate, followed by ‘Q’ (Queenstown), and ‘S’ (Southampton)
* By Cabin Deck: There are clear differences in survival rates between decks, with some upper decks having higher survival probabilities
* By Family Size: Passengers traveling in small to medium-sized family groups (2-4 members) generally had higher survival probabilities than those traveling alone or in very large family groups