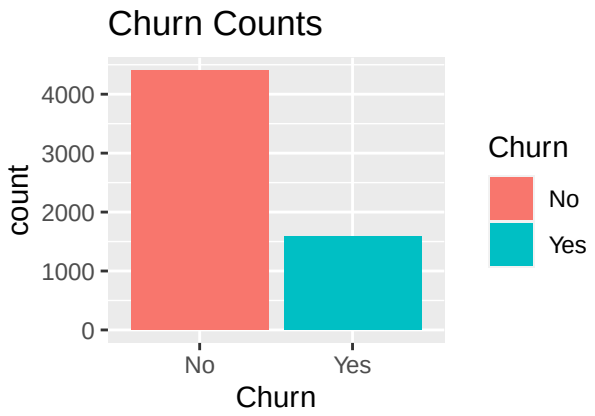
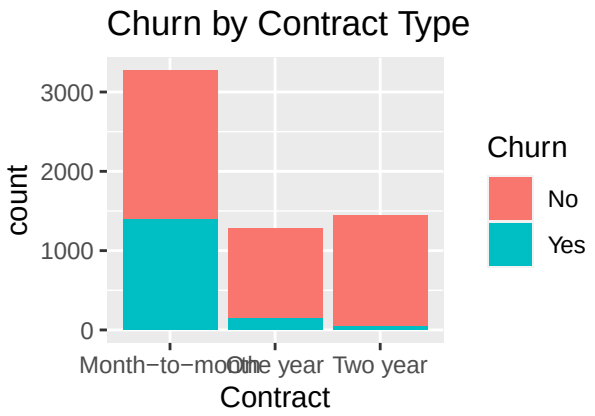
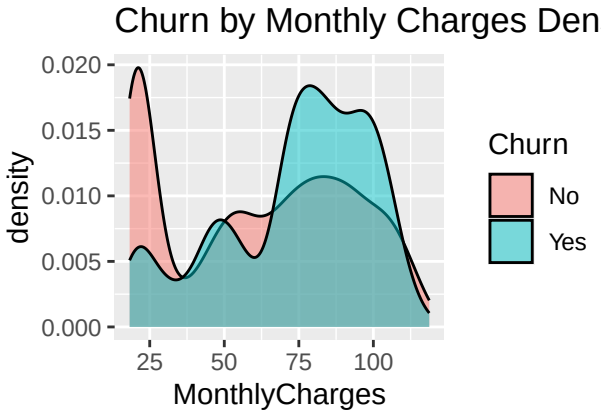
report.r

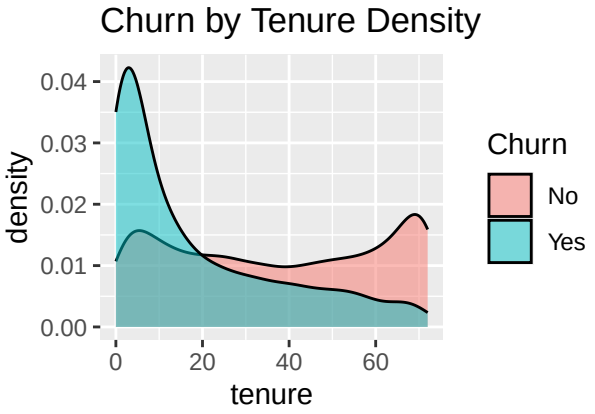
```r
library(ggplot2)
library(dplyr)
library(readr)
library(readr)
data <- read_csv("telecom.csv")
```

```r
library(ggplot2)
library(gridExtra)
plot1 <- ggplot(data, aes(x = Churn, fill = Churn)) +
  geom_bar() +
  labs(title = "Churn Counts")
plot2 <- ggplot(data, aes(x = Contract, fill = Churn)) +
  geom_bar() +
  labs(title = "Churn by Contract Type")
plot3 <- ggplot(data, aes(x = MonthlyCharges, fill = Churn)) +
  geom_density(alpha = 0.5) +
  labs(title = "Churn by Monthly Charges Density")
plot4 <- ggplot(data, aes(x = tenure, fill = Churn)) +
  geom_density(alpha = 0.5) +
  labs(title = "Churn by Tenure Density")
grid.arrange(plot1, plot2, plot3, plot4, nrow = 2, ncol = 2)
```










```
library(dplyr)
library(readr)
library(tidyr)
library(caret)
data$TotalCharges <- ifelse(data$TotalCharges == " ", NA, as.numeric(data$TotalCharges))
data <- na.omit(data)
data <- data %>%mutate_if(is.character, as.factor)
set.seed(123)
trainIndex <- createDataPartition(data$Churn, p = 0.7, list = FALSE)
trainData <- data[trainIndex, ]
testData <- data[-trainIndex, ]
model <- glm(Churn ~ ., data = trainData, family = "binomial")
predictions <- predict(model, newdata = testData, type = "response")
predicted_classes <- ifelse(predictions > 0.5, 1, 0)
ljhgmatrix <- table(testData$Churn, predicted_classes)
ljhgmatrix
```

```
##       predicted_classes
##           0    1
##   No   1187  129
##   Yes   232  244
```

```
set.seed(123)
sjsl <- train(
  Churn ~ .,
```

```
  data = trainData,
  method = "rf",
  trControl = trainControl(method = "cv", number = 5),
  tuneLength = 5
)
sjsl$results
```

```
##   mtry  Accuracy     Kappa  AccuracySD    KappaSD
## 1    2 0.7923064 0.3591292 0.009869159 0.02689659
## 2    9 0.7937426 0.4269549 0.014005680 0.04209284
## 3   16 0.7913548 0.4225485 0.013442456 0.03893407
## 4   23 0.7892074 0.4216887 0.018699639 0.04977653
## 5   30 0.7875348 0.4176068 0.020765950 0.05497415
```

```
bestmodel <- sjsl$bestTune
bestmodel
```

```
##   mtry
## 2    9
```

```
sjslpredictions <- predict(sjsl, newdata = testData)
sjslmatrix <- table(testData$Churn, sjslpredictions)
sjslmatrix
```

```
##      sjslpredictions
##         No  Yes
##   No  1197  119
##   Yes  256  220
```

```
sjslaccuracy <- sum(diag(sjslmatrix)) / sum(sjslmatrix)
sjslaccuracy
```

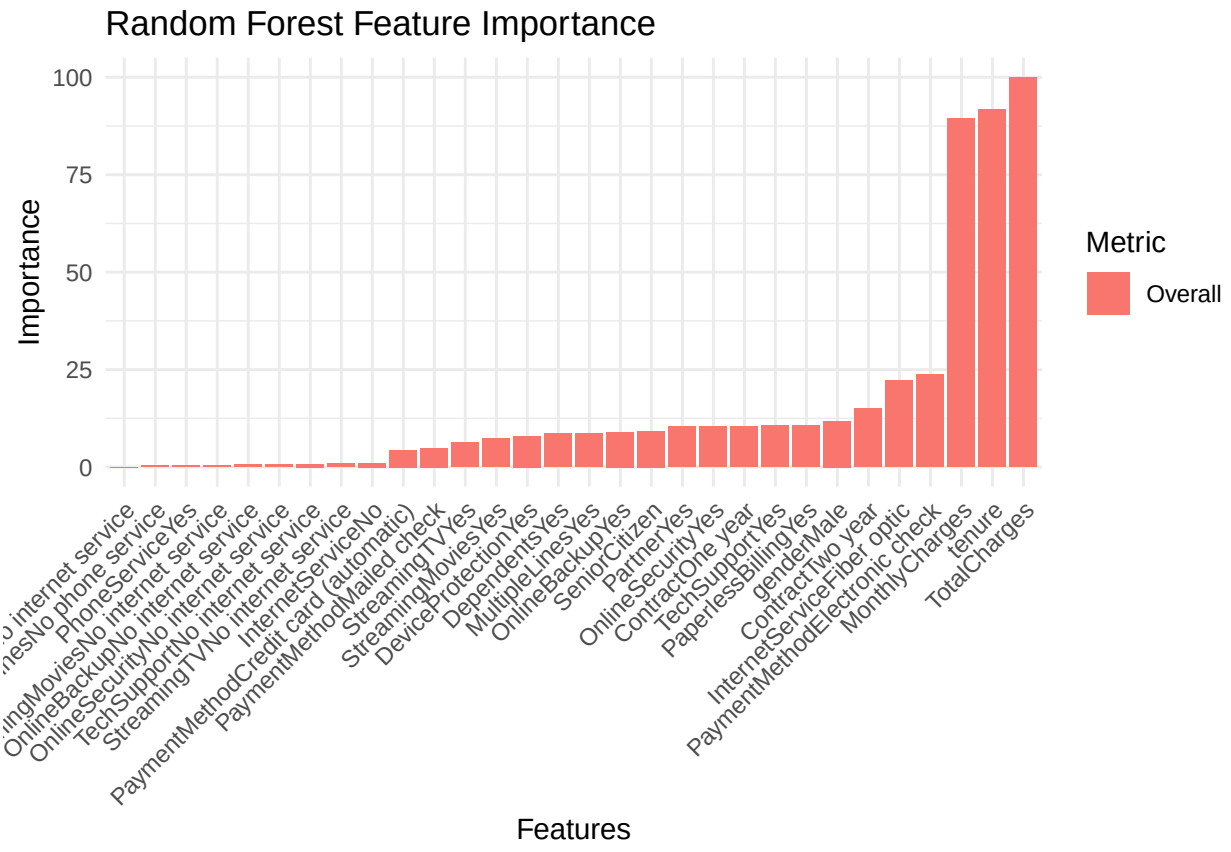
```
## [1] 0.7907366
```

```
sjslimportance <- varImp(sjsl)
library(ggplot2)
library(tidyr)

sjslimportancedf <- as.data.frame(sjslimportance$importance)
sjslimportancedf$Features <- rownames(sjslimportance$importance)
sjslimportancedf <- gather(sjslimportancedf, key = "Metric", value = "Value", -Features)

importanceplot <- ggplot(sjslimportancedf, aes(x = reorder(Features, Value), y = Value, fill = Metric)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Random Forest Feature Importance",
       x = "Features",
       y = "Importance") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

print(importanceplot)
```



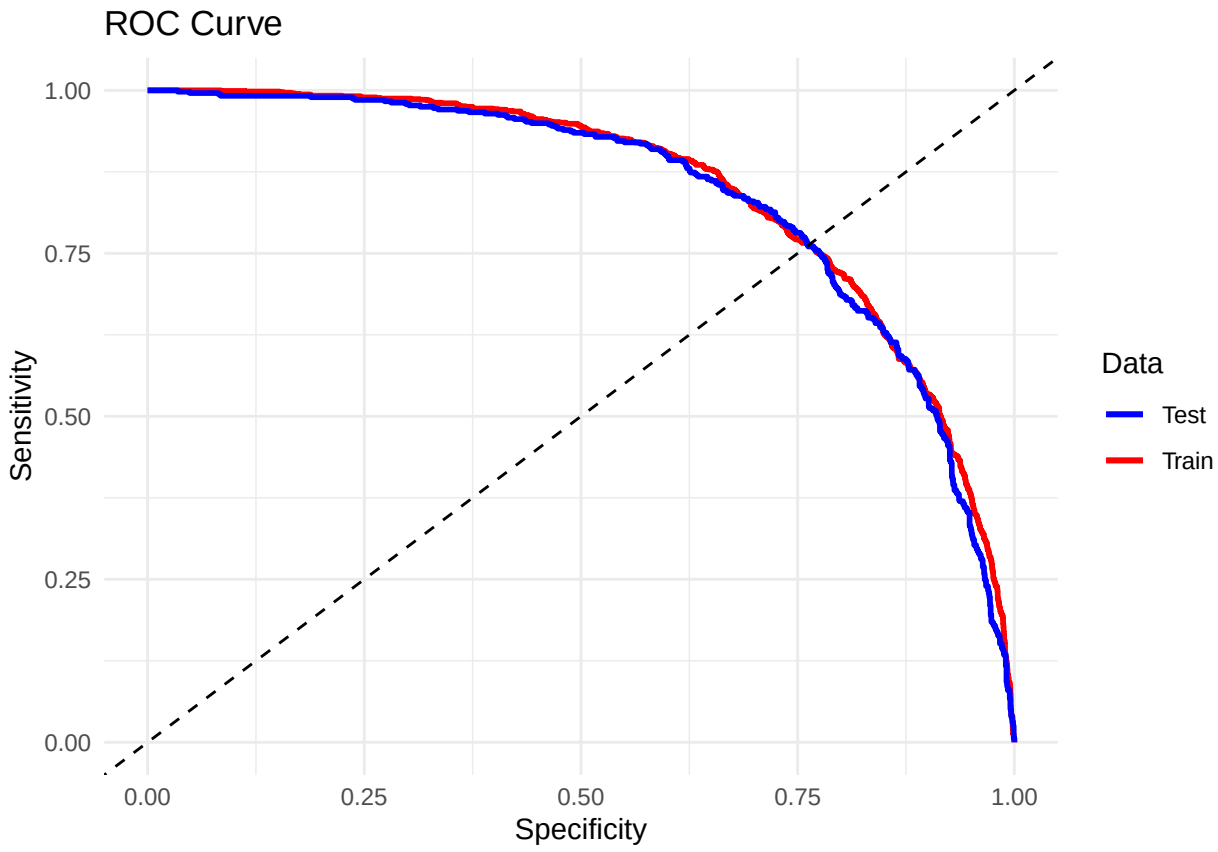

```
library(pROC)
train_predictions <- predict(model, newdata = trainData, type = "response")
train_roc <- roc(trainData$Churn, train_predictions)
test_roc <- roc(testData$Churn, predictions)
roc_plot <- ggplot() +
  geom_line(aes(x = train_roc$specificities, y = train_roc$sensitivities, color = "Train"), size = 1) +
  geom_line(aes(x = test_roc$specificities, y = test_roc$sensitivities, color = "Test"), size = 1) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "black") +
  scale_color_manual(values = c("blue", "red")) +
  labs(x = "Specificity", y = "Sensitivity", color = "Data") +
  ggtitle("ROC Curve") +
  theme_minimal()
print(paste("Train AUC:", auc(train_roc)))
```

```
## [1] "Train AUC: 0.847546192953171"
```

```
print(paste("Test AUC:", auc(test_roc)))
```

```
## [1] "Test AUC: 0.838793230057981"
```

```
print(roc_plot)
```


ROC Curve
1.00
0.75
0.50
0.25
0.00
Sensitivity
0.00
0.25
0.50
0.75
1.00
Specificity
Data
Test
Train