Titanic: Machine Learning from Disaster

逍遥子晴

2017 年 8 月 26 日

解答本题共分为四部分：

1、载入数据，数据理解。

2、探究 Surived 变量与其它各变量是否相关（本文只孤立的探究了单个变量与 Surived 的关系，未涉及变量组合后的相关性探究）这一步也称作特征工程，关于特征工程的知识还未仔细研究。

3、数据预处理，主要是缺失数据的处理。

4、根据步骤 2，找出相关变量，用随机森林算法对测试数据进行预测。

导入所需要的包。

```
> # install.packages('ggthemes')
> library("ggthemes")  #ggplot2 主题扩展包
> library("ggplot2")
> library("dplyr")  # 数据处理
```

```
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("mice")  # 用来处理缺失数据
> library("randomForest")  # 随机森林算法
```

```
randomForest 4.6-12

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("rpart")
```

一、数据理解

1、载入训练数据和测试数据，为了后续处理方便，将两组数据进行合并。查看数据中整体结构。

```
> ## 熟悉数据整体情况
> train <- read.csv("train.csv")
> test <- read.csv("test.csv")
> data <- bind_rows(train, test)
> str(data)
```

```
'data.frame':   1309 obs. of  12 variables:
 $ PassengerId: int  1 2 3 4 5 6 7 8 9 10 ...
 $ Survived   : int  0 1 1 1 0 0 0 0 1 1 ...
 $ Pclass     : int  3 1 3 1 3 3 1 3 3 2 ...
 $ Name      : chr  "Braund, Mr. Owen Harris" "Cumings, Mrs. John Bradley (Florence Briggs Thayer)" "Heikkinen
 $ Sex        : Factor w/ 2 levels "female","male": 2 1 1 1 2 2 2 2 1 1 ...
 $ Age        : num  22 38 26 35 35 NA 54 2 27 14 ...
 $ SibSp      : int  1 1 0 1 0 0 0 3 0 1 ...
 $ Parch      : int  0 0 0 0 0 0 0 1 2 0 ...
 $ Ticket     : chr  "A/5 21171" "PC 17599" "STON/O2. 3101282" "113803" ...
 $ Fare       : num  7.25 71.28 7.92 53.1 8.05 ...
 $ Cabin      : chr  "" "C85" "" "C123" ...
 $ Embarked   : chr  "S" "C" "S" "S" ...
```

数据中共有 1309 行，包含 12 个变量，其中 PassengerId 可忽略，其余变量含义如下：

- Survived：生存情况，1 为存活，0 为死亡
- Pclass：客舱等级，1 为高级，2 为中级，3 为低级
- Name：乘客名字
- Sex：乘客性别
- Age：乘客年龄
- SibSp：在船兄弟姐妹数/配偶数
- Parch：在船父母数/子女数
- Ticket：船票编号
- Fare：船票价格
- Cabin：客舱号
- Embarked：登船港口

2、判断数据中是否存在缺失值。

```r
> # 判断数据中是否存在缺失值（NA 和空值）
> sapply(data, function(x) sum(is.na(x)))  # 判断数值型数据
```

```
PassengerId    Survived      Pclass        Name         Sex         Age
          0         418           0           0           0         263
      SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0           1           0           0
```

```r
> sapply(data, function(x) sum(x == ""))  # 判断类别性数据
```

```
PassengerId    Survived      Pclass        Name         Sex         Age
          0          NA           0           0           0          NA
      SibSp       Parch      Ticket        Fare       Cabin    Embarked
          0           0           0          NA        1014           2
```

由输出结果可知，缺失值情况如下：

- Survived 有 418 个缺失值是由于测试集的原因，可忽略。
- Fare 有 1 个缺失值
- Age 有 263 个缺失值
- Cabin 有 1014 个缺失值
- Embarked 有 2 个缺失值

因缺失值数量较大，先对数据进行分析，回过头再进行缺失值处理。

二、各变量与 Survived 变量相关性探究。

针对类别型变量，采用做柱状图形式直观的查看相关信息，代码类似。针对数值型数据采用线性做题进行探究。

```r
> # Survived 变量因子化
> data$Survived <- factor(data$Survived)
```

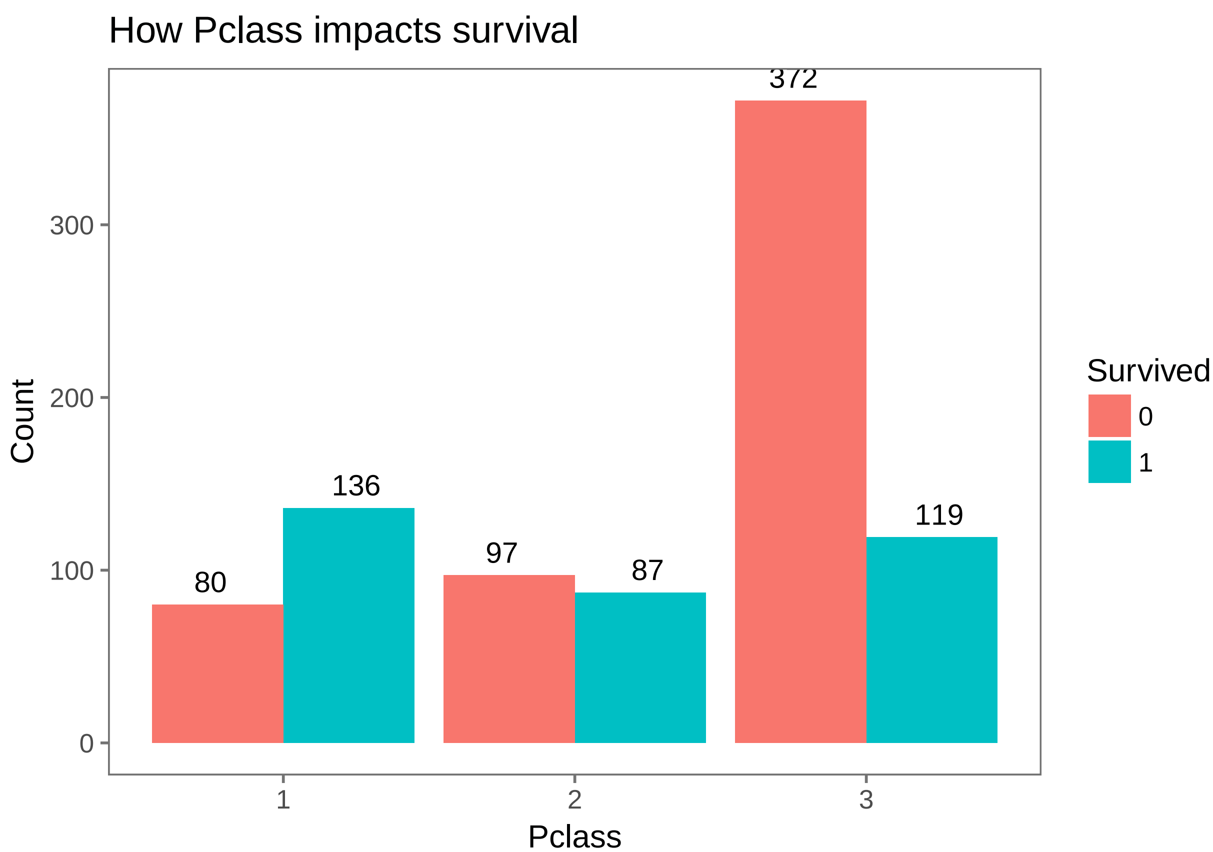
（1）PClass 变量与 Survived 的关系

```r
> ### 探究幸存率与各个变量的关系 PClass 变量与 Survived 的关系
> data$Pclass <- factor(data$Pclass)  # 因子化
> prop.table(table(data$Pclass, data$Survived), 1)  # 计算各等级客舱的幸存率
```

```

          0         1
  1 0.3703704 0.6296296
  2 0.5271739 0.4728261
  3 0.7576375 0.2423625
```

```
> ggplot(data = data[1:nrow(train), ], mapping = aes(x = Pclass, fill = Survived)) +
+     geom_bar(stat = "count", position = "dodge") + xlab("Pclass") + ylab("Count") +
+     ggtitle("How Pclass impacts survival") + geom_text(stat = "count", aes(label = ..count..),
+     position = position_dodge(width = 1), vjust = -0.6) + theme_few()
```

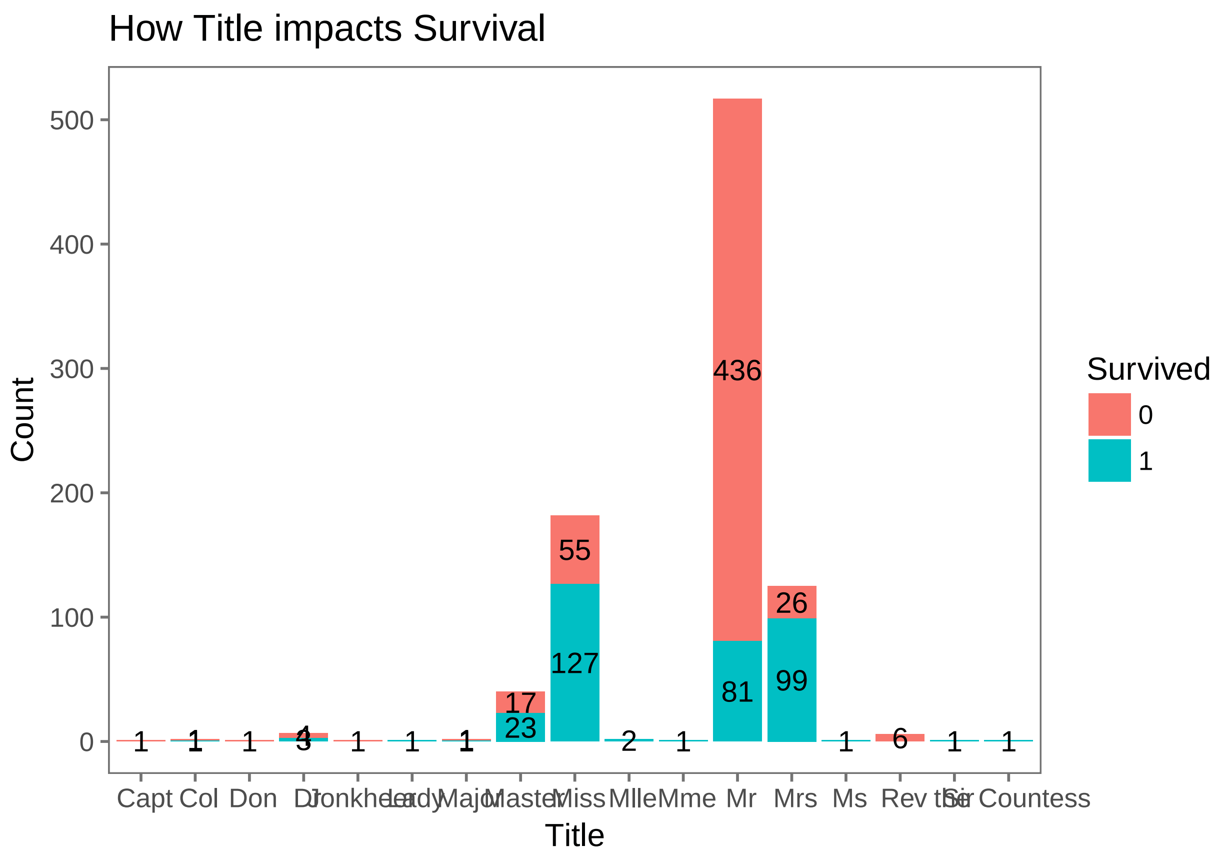


Pclass 为 1 的幸存率最高，Pclass 为 3 的幸存率最低。

（2）根据 Name 变量提取出部分信息，增加 Title 变量。

```
> # 将 Name 变量中有关 Title 的信息抽取出来
> data$Name <- as.character(data$Name)
> data$Title <- sapply(data$Name, FUN = function(x) {
+     strsplit(x, split = "[,.]")[[1]][2]
+ })
> # 将出现次数较少的类别归为一类
> data$Title[data$Title %in% c("Mme", "Mlle")] <- "Mlle"
> data$Title[data$Title %in% c("Capt", "Don", "Major", "Sir")] <- "Sir"
> data$Title[data$Title %in% c("Dona", "Lady", "the Countess", "Jonkheer")] <- "Lady"
> data$Title <- factor(data$Title)
> ggplot(data = data[1:nrow(train), ], mapping = aes(x = Title, y = ..count..,
+     fill = Survived)) + geom_bar(stat = "count", position = "stack") + xlab("Title") +
+     ylab("Count") + ggtitle("How Title impacts Survival") + geom_text(stat = "Count",
```

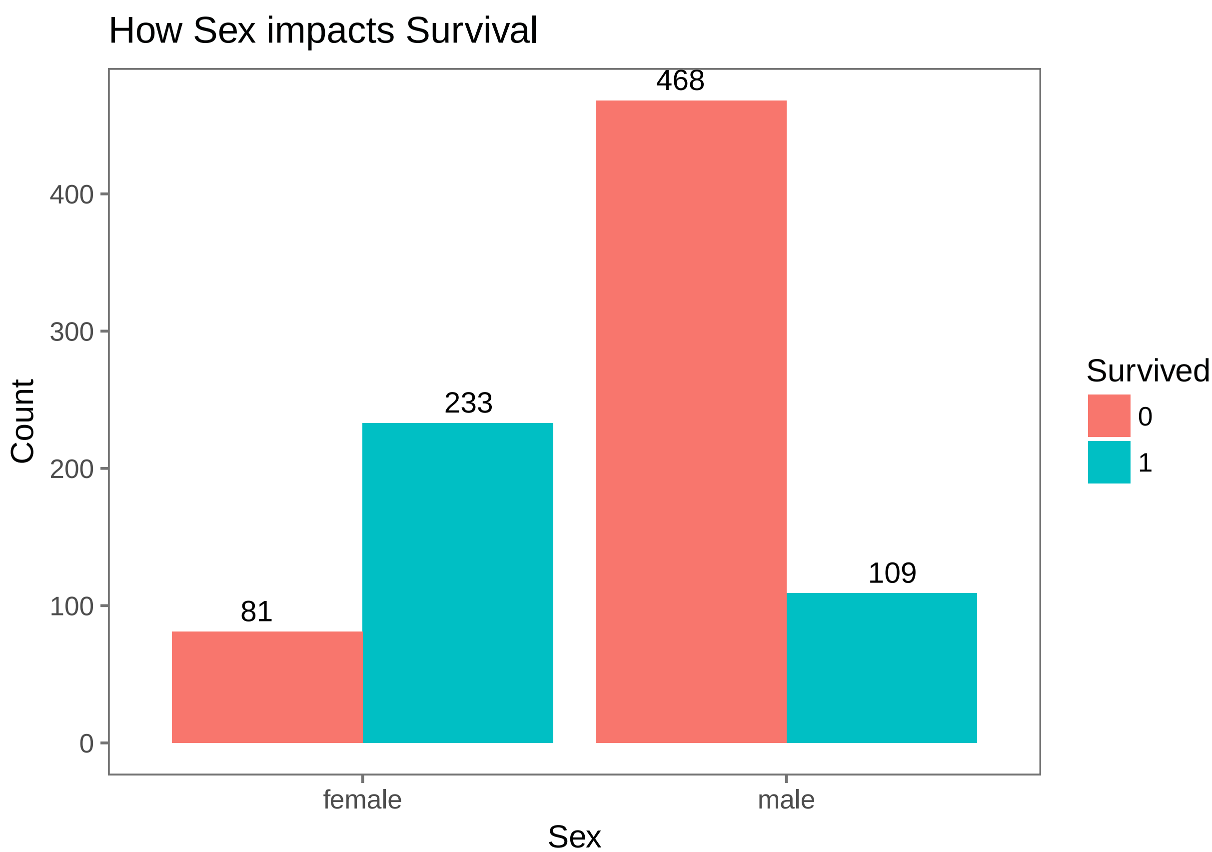
```
+     aes(label = ..count..), position = position_stack(vjust = 0.5)) + theme_few()
```



Title 为 Mrs 和 Miss 的幸存率比较大，为 Mr 的幸存率比较小。

（3）Sex 变量与 Survived 的关系

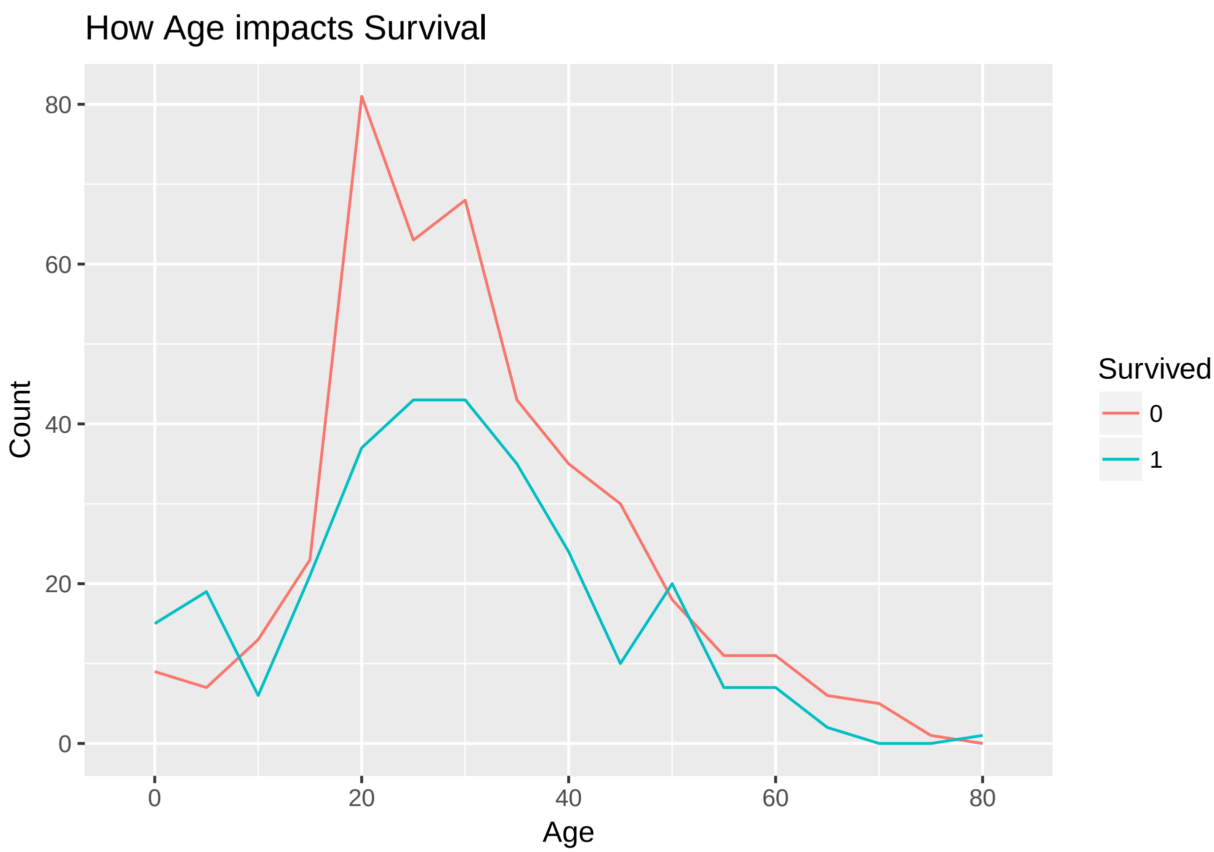
```
> # Sex 变量与 Survived 的关系
> data$Sex <- factor(data$Sex)
> ggplot(data = data[1:nrow(train), ], mapping = aes(x = Sex, y = ..count.., fill = Survived)) +
+     geom_bar(stat = "count", position = "dodge") + xlab("Sex") + ylab("Count") +
+     ggtitle("How Sex impacts Survival") + geom_text(stat = "count", aes(label = ..count..),
+     position = position_dodge(width = 1), vjust = -0.5) + theme_few()
```



女性的幸存率更高。

（4）Age 变量与 Survived 的关系

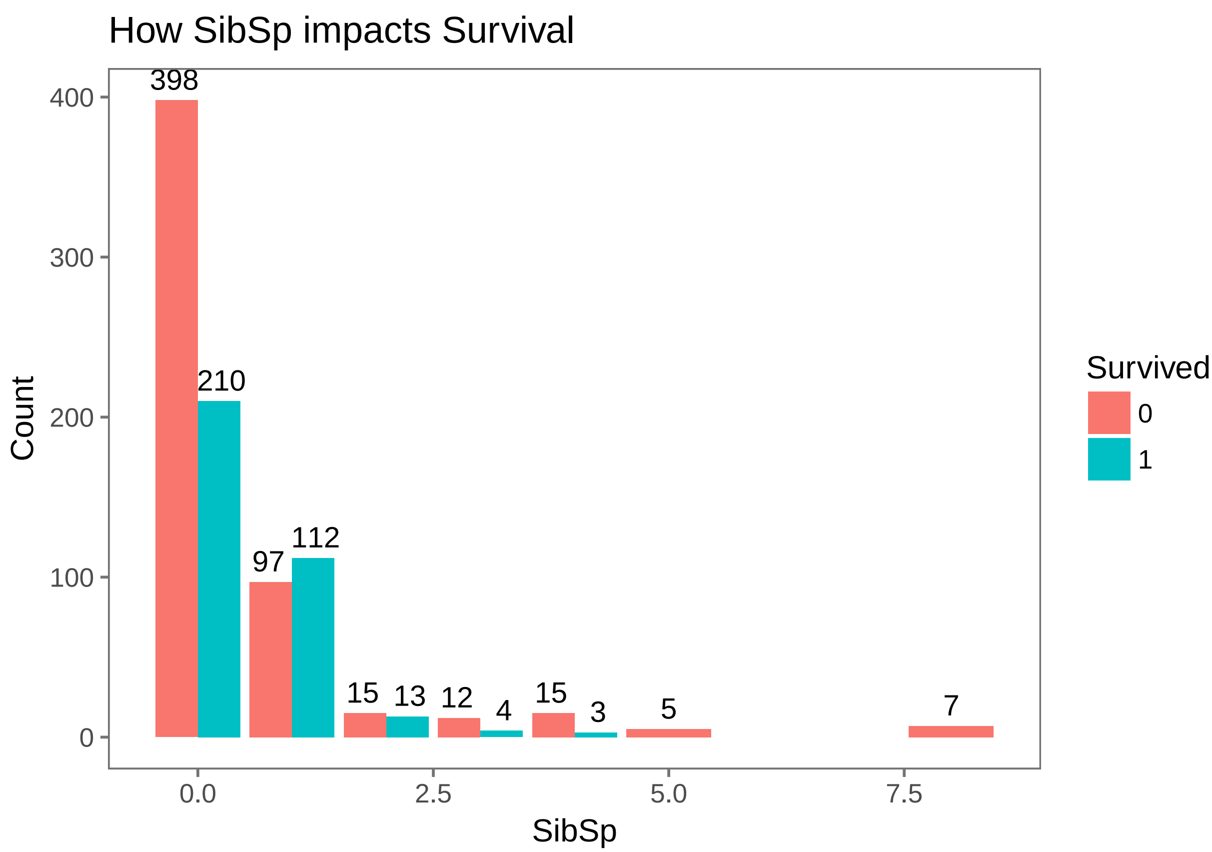
```
> # Age 变量与 Survived 的关系
> ggplot(data = data[(!is.na(data$Age)) & row(as.matrix(data[, "Age"])) <= 891,
+     ], aes(x = Age, color = Survived)) + geom_line(aes(label = ..count..), stat = "bin",
+     binwidth = 5, na.rm = TRUE) + labs(title = "How Age impacts Survival", x = "Age",
+     y = "Count", fill = "Survived")
```



未成年人的幸存率更高。

（5）SibSp 变量与 Survived 的关系

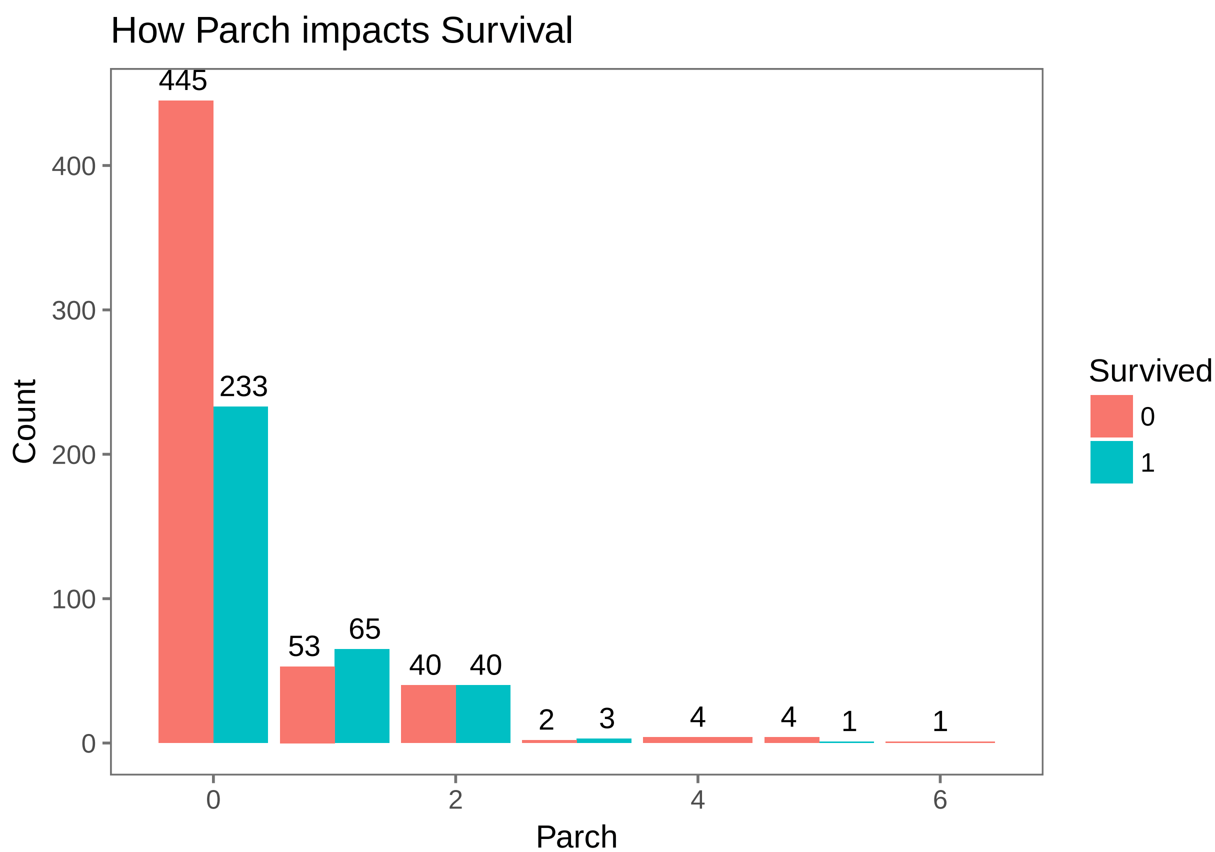
```
> # SibSp 变量与 Survived 的关系
> ggplot(data = data[1:nrow(train), ], mapping = aes(x = SibSp, y = ..count..,
+     fill = Survived)) + geom_bar(stat = "count", position = "dodge") + xlab("SibSp") +
+     ylab("Count") + ggtitle("How SibSp impacts Survival") + geom_text(stat = "count",
+     aes(label = ..count..), position = position_dodge(width = 1), vjust = -0.5) +
+     theme_few()
```



SibSp 为 1 或 2 的乘客幸存率最高。

（6）Parch 变量与 Survived 的关系

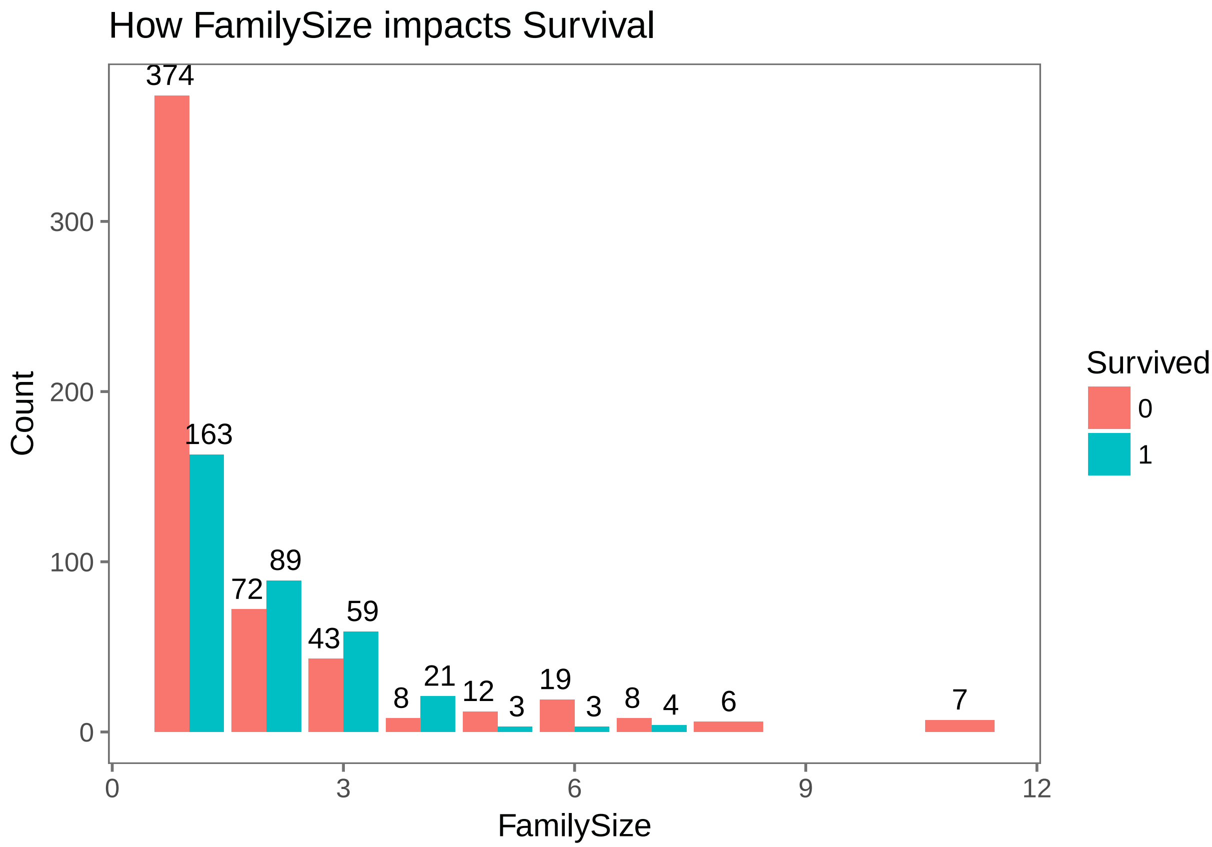
```
> # Parch 变量与 Survived 的关系
>
> ggplot(data = data[1:nrow(train), ], mapping = aes(x = Parch, y = ..count..,
+     fill = Survived)) + geom_bar(stat = "count", position = "dodge") + xlab("Parch") +
+     ylab("Count") + ggtitle("How Parch impacts Survival") + geom_text(stat = "count",
+     aes(label = ..count..), position = position_dodge(width = 1), vjust = -0.5) +
+     theme_few()
```



SibSp 为 1 或 2 的乘客幸存率最高。

（7）根据 Parch 变量与 SibSp 变量计算家庭成员数量，并生成新的变量 FamilySize。

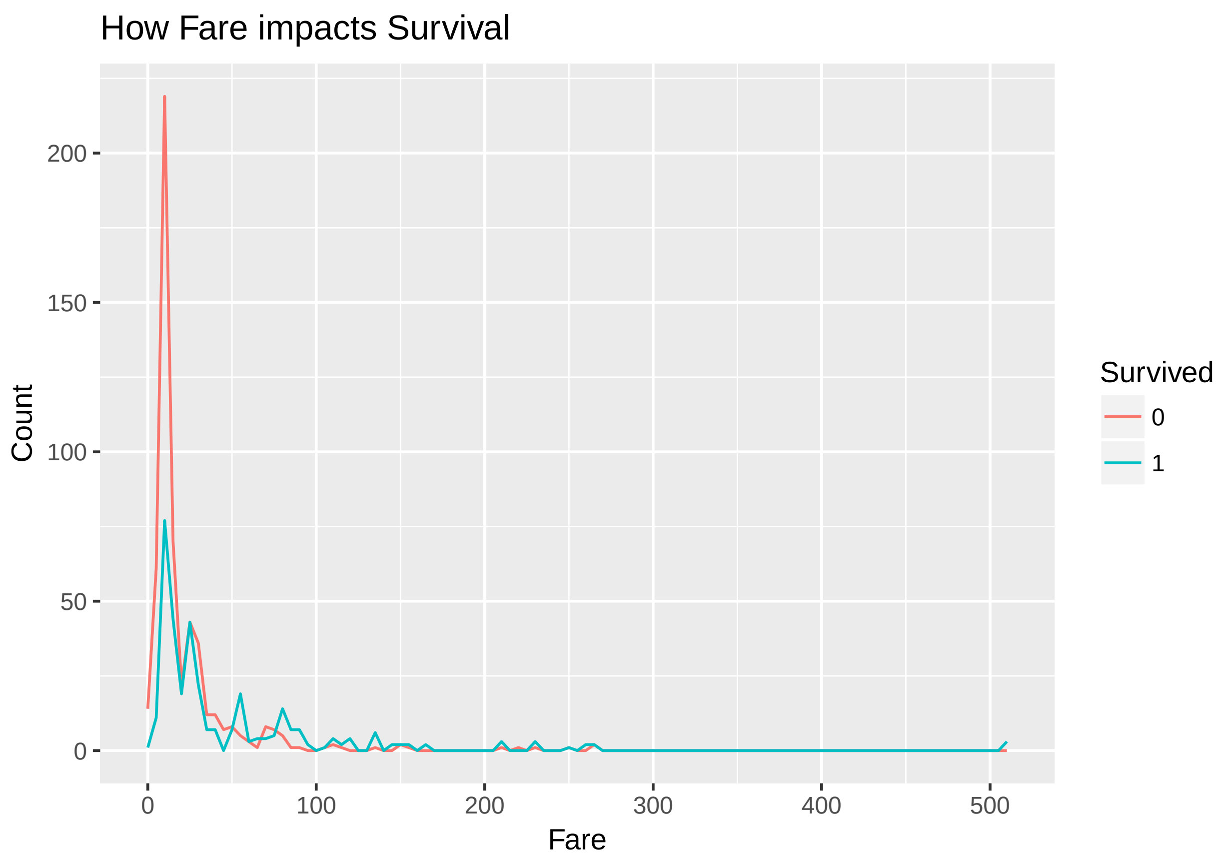
```
> # 新增 FamilySize 变量，探索与 Survived 的关系
> data$FamilySize <- data$Parch + data$SibSp + 1
> ggplot(data = data[1:nrow(train), ], mapping = aes(x = FamilySize, y = ..count..,
+     fill = Survived)) + geom_bar(stat = "count", position = "dodge") + xlab("FamilySize") +
+     ylab("Count") + ggtitle("How FamilySize impacts Survival") + geom_text(stat = "count",
+     aes(label = ..count..), position = position_dodge(width = 1), vjust = -0.5) +
+     theme_few()
```



FamilySize 为 2 到 4 的乘客幸存率最高。

（8）Fare 变量与 Survived 的关系

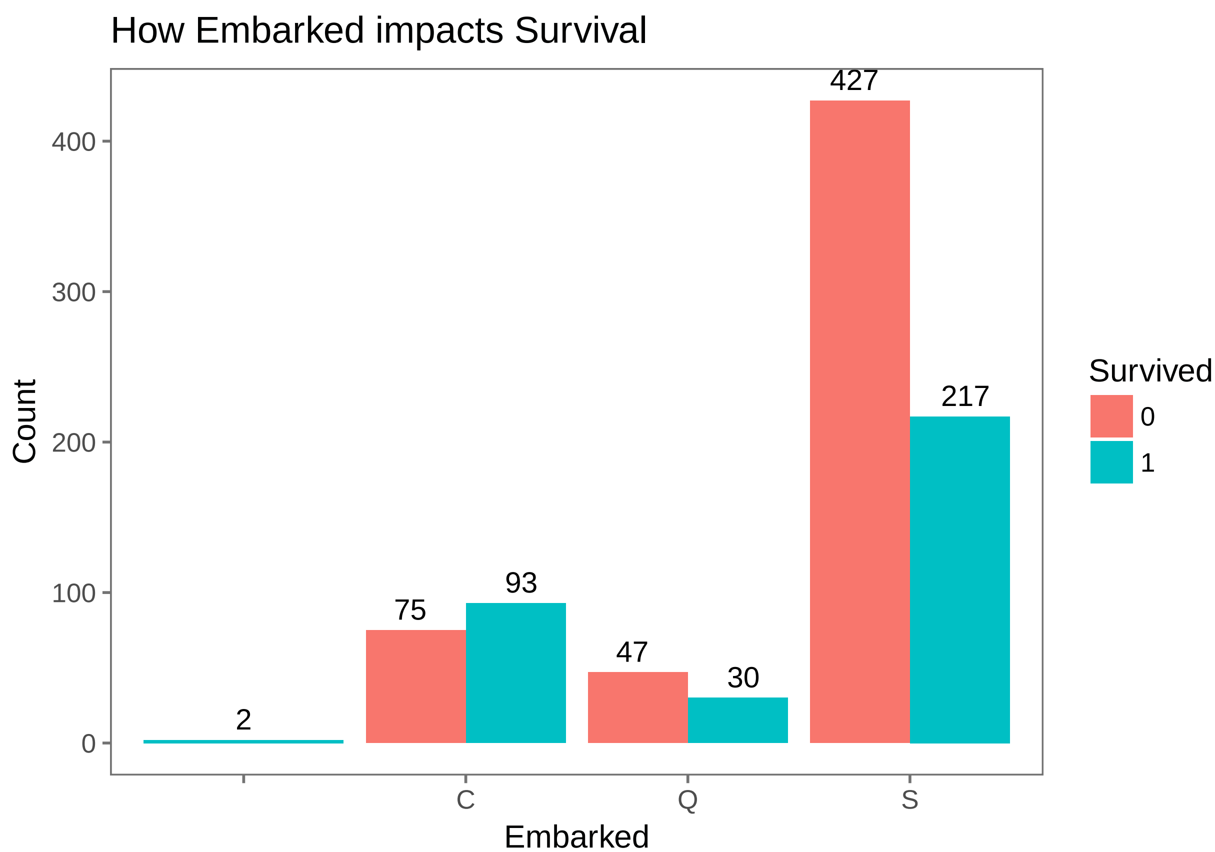
```
> # Fare 变量与 Survived 的关系
> ggplot(data = data[(!is.na(data$Fare)) & row(as.matrix(data[, "Fare"])) <= 891,
+     ], aes(x = Fare, color = Survived)) + geom_line(aes(label = ..count..),
+     stat = "bin", binwidth = 5, na.rm = TRUE) + labs(title = "How Fare impacts Survival",
+     x = "Fare", y = "Count", fill = "Survived")
```



票价越高幸存率越高。

（9）Embarked 变与 Survived 的关系

```
> # Embarked 变量与 Survived 的关系
>
> ggplot(data = data[1:nrow(train), ], mapping = aes(x = Embarked, y = ..count..,
+     fill = Survived)) + geom_bar(stat = "count", position = "dodge") + xlab("Embarked") +
+     ylab("Count") + ggtitle("How Embarked impacts Survival") + geom_text(stat = "count",
+     aes(label = ..count..), position = position_dodge(width = 1), vjust = -0.5) +
+     theme_few()
```



Embarked 为 C 或 NA 的幸存率最高。

三、缺失值处理

1、Fare 代表票价，为数值型数据，有 1 个缺失值。采取中位数填补法。

```
data$Fare[is.na(data$Fare)]<-median(data$Fare,na.rm = TRUE)
```

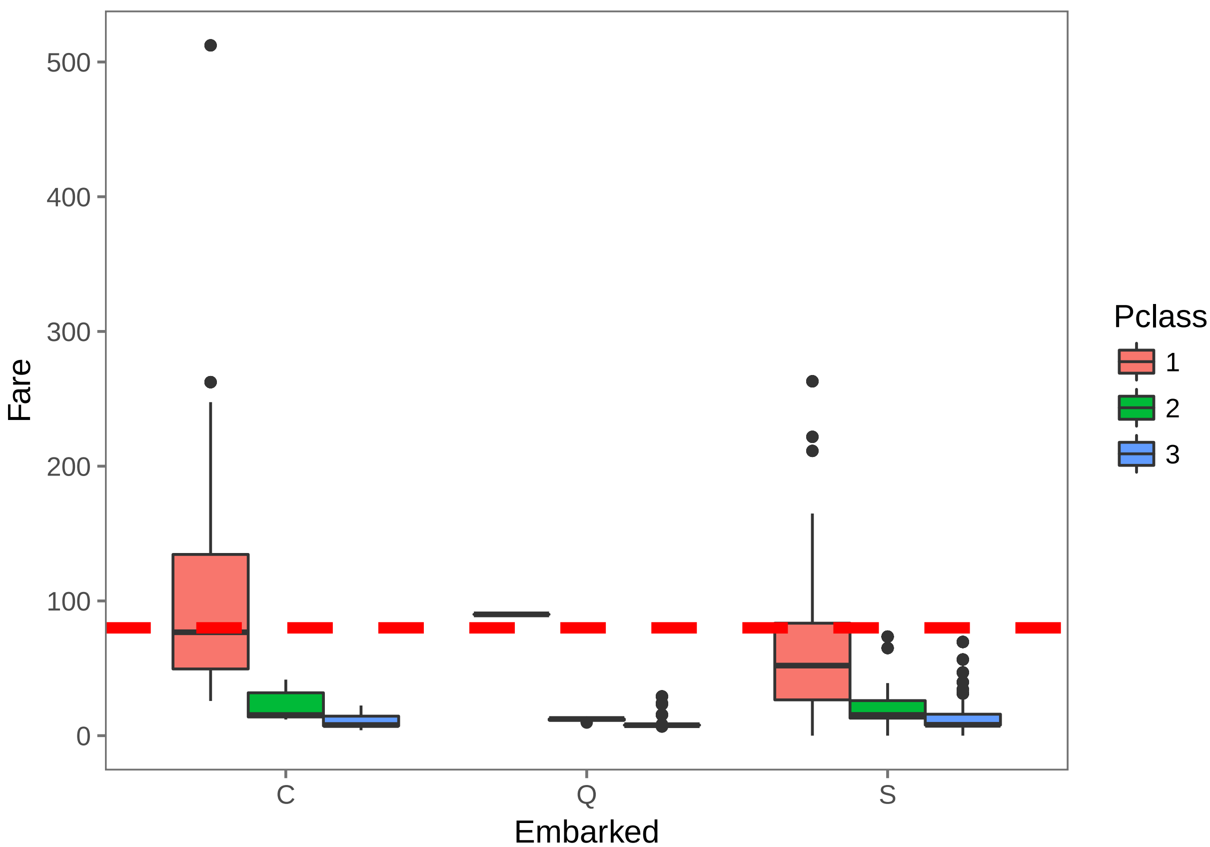
2、Embarked 有 2 个缺失值，先将这两个缺失值对应的乘客信息选取出来

```
> data[data$Embarked == "", c("PassengerId", "Pclass", "Fare", "Embarked")]
```

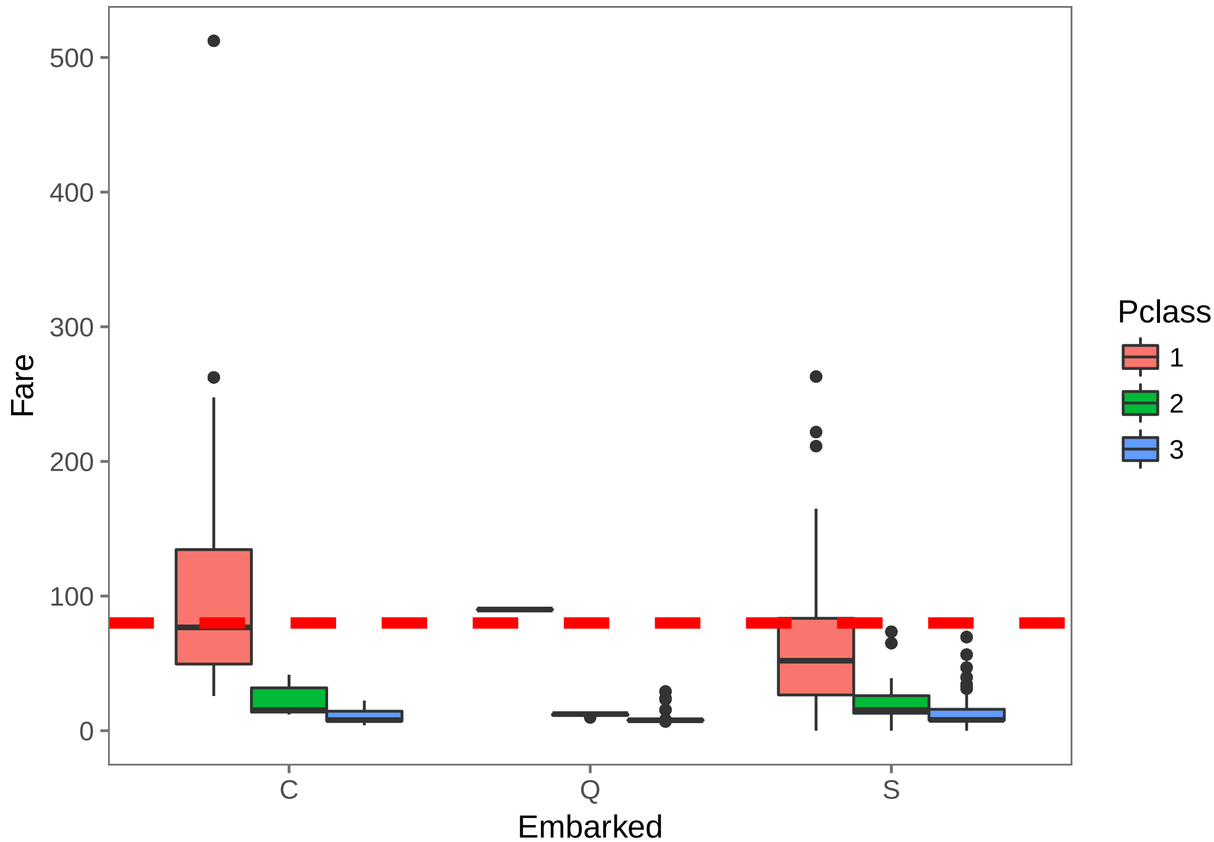
```
    PassengerId Pclass Fare Embarked
62           62      1   80
830         830      1   80
```

输出结果可知，Pclass 都为 1，Fare 都为 80。

```
> ggplot(data = data[data$Embarked != "", ], aes(x = Embarked, y = Fare, fill = Pclass)) +
+     geom_boxplot() + geom_hline(aes(yintercept = 80), color = "red", linetype = "dashed",
+     lwd = 2) + theme_few()
```




```
> ggplot(data = data[data$Embarked != "", ], aes(x = Embarked, y = Fare, fill = Pclass)) +
+     geom_boxplot() + geom_hline(aes(yintercept = 80), color = "red", linetype = "dashed",
+     lwd = 2) + theme_few()
```

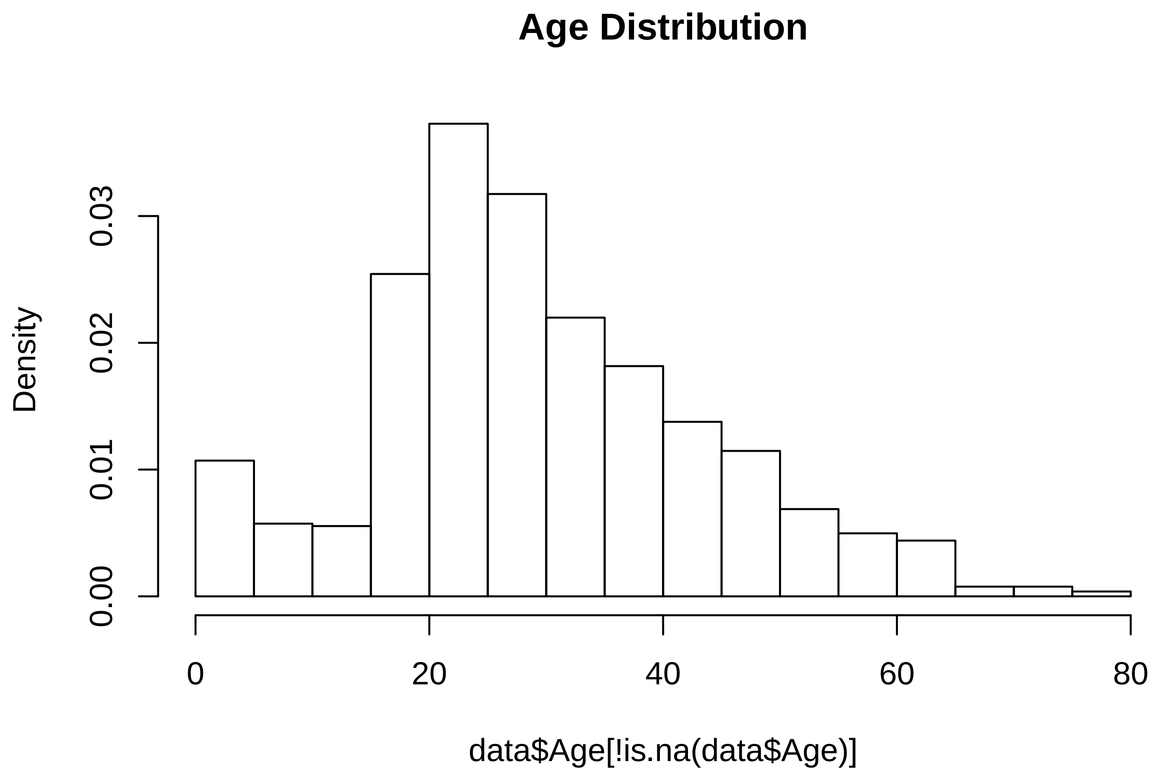


Embarked 为 C 的 Pclass 属于 1 的 Fare 中位数正好是 80，所以将缺失值填补为 C。

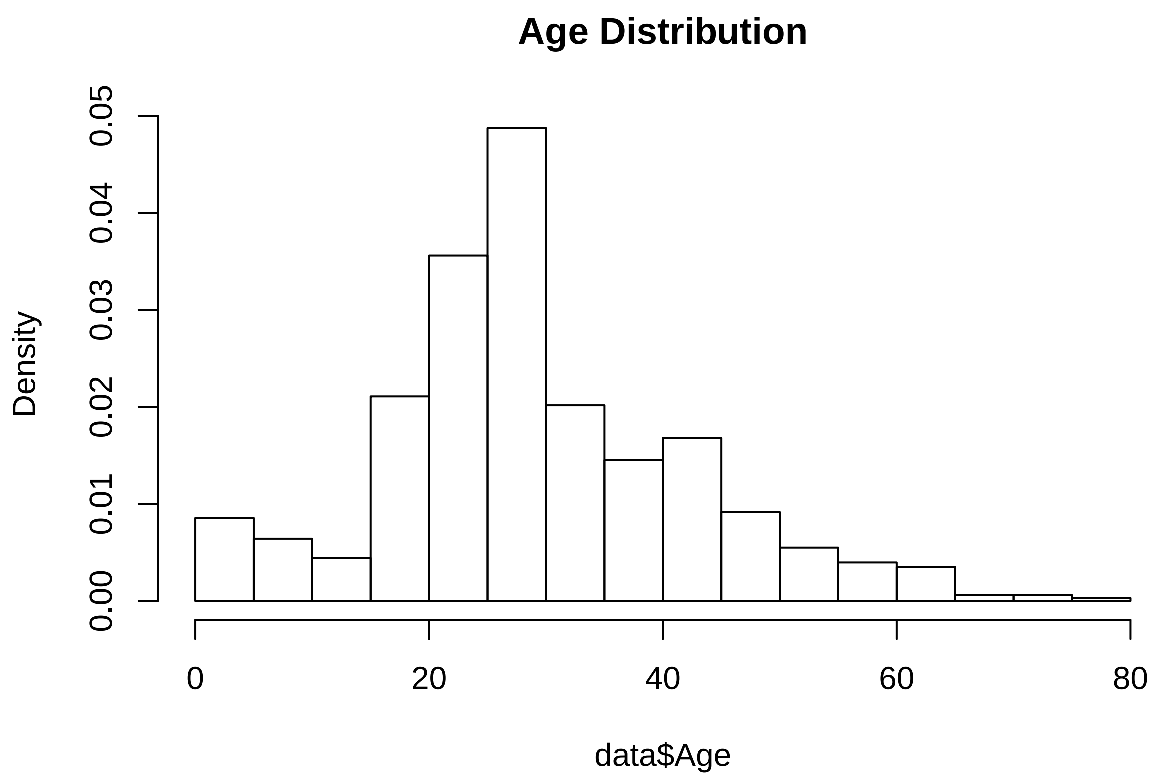
```
> data$Embarked[c(62, 830)] <- "C"
> data$Embarked <- factor(data$Embarked)
```

3、预测填补 Age 的缺失值，用到了决策树方法。

```
> hist(data$Age[!is.na(data$Age)], freq = F, main = "Age Distribution")
```




```
> age.model <- rpart(Age ~ Pclass + Sex + SibSp + Parch + Fare + Embarked + Title +
+     FamilySize, data = data[!is.na(data$Age), ], method = "anova")
> data$Age[is.na(data$Age)] <- predict(age.model, data[is.na(data$Age), ])
> hist(data$Age, freq = F, main = "Age Distribution")
```



根据缺失值填补前后年龄的分布情况可知，数据填补是合理的。

4、由于 Cabin（客舱号）数据缺失量较大，这里暂不考虑作为相关性变量。

四、构建模型，预测数据。

根据第三步的分析，我们锁定了 9 个与 Survived 相关的变量，分别为

```
> train <- data[1:891, ]
> test <- data[892:1309, ]
> set.seed(754)
> # 构建预测模型
> rf_model <- randomForest(factor(Survived) ~ Pclass + Sex + Age + Fare + Embarked +
+     Title + FamilySize + Embarked + SibSp, data = train)
> prediction <- predict(rf_model, test)
> # 保存数据结果 passagerId 和 survived 参数
> solution <- data.frame(PassengerID = test$PassengerId, Survived = prediction)
> # 保存到文件
> write.csv(solution, file = "predict_Solution.csv", row.names = F)
```

结果上传后排名 2768，0.78947。