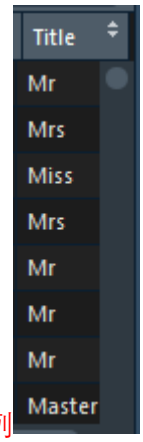
###1

```
train <- read.csv("train.csv",stringsAsFactors = F)     #读取数据，不作为因子
test <- read.csv("test.csv",stringsAsFactors = F)
full   <- bind_rows(train, test)                        # bind training & test data 包 dplyr
str(full)                                               #列举信息结构
full$Title <- gsub('(.*, )|(\\..*)', '', full$Name)     #。代表一个字符，*代表重复很多次，\\.输入一个点  .*后面的字符
```



#增加一列

```
table(full$Sex, full$Title)                             #左 sex，右 title
```

```
> table(full$Sex, full$Title)

        Capt Col Don Dona  Dr Jonkheer Lady Major Master Miss
  female    0   0   0    1   1        0    1     0      0  260
  male      1   4   1    0   7        1    0     2     61    0

        Mlle Mme  Mr Mrs  Ms Rev Sir the Countess
  female    2   1   0 197   2   0   0            1
  male      0   0 757   0   0   8   1            0
>
```

```
rare_title <- c('Dona', 'Lady', 'the Countess','Capt', 'Col', 'Don',
                'Dr', 'Major', 'Rev', 'Sir', 'Jonkheer')    #数量少的合并
full$Title[full$Title == 'Mlle']        <- 'Miss'
full$Title[full$Title == 'Ms']          <- 'Miss'
full$Title[full$Title == 'Mme']         <- 'Mrs'
full$Title[full$Title %in% rare_title]  <- 'Rare Title'      #选出 rare
table(full$Sex, full$Title)
```

```
> table(full$Sex, full$Title)

        Master Miss  Mr Mrs Rare Title
  female      0  264   0 198          4
  male       61    0 757   0         25
```

```
full$Surname <- sapply(full$Name,
                       function(x) strsplit(x, split = '[,.]')[[1]][1])
```

#对 Name 执行函数，strsplit，分隔符是，。并选取 list 的 1 组，再选第一个

Surname
Braund
Cumings
Heikkinen
Futrelle
Allen
Moran
McCarthy
Palsson

增加一列

```
cat(paste('We have <b>', nlevels(factor(full$Surname)), '</b> unique surnames. I would be interested to infer ethnicity based on surname --- another time.'))
#875 unique surnames

###2
full$Fsize <- full$SibSp + full$Parch + 1        #确定家庭大小
```

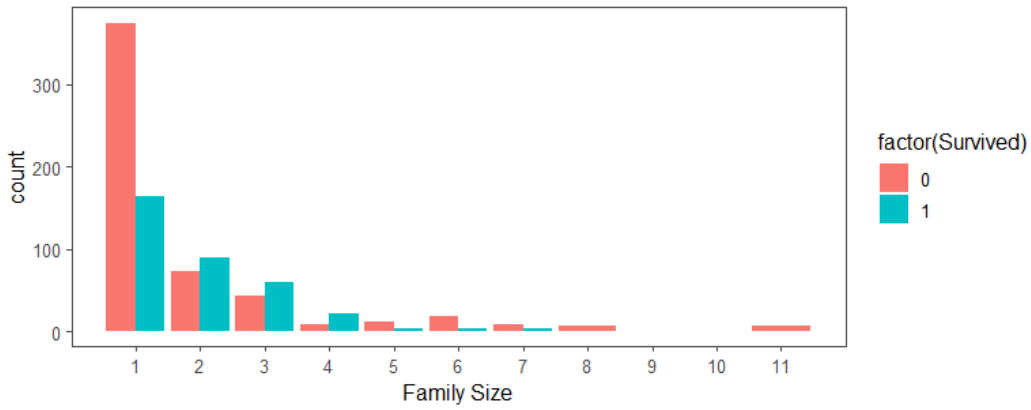
Fsize
2
2
1
2
1
1
1
5

```
full$Family <- paste(full$Surname, full$Fsize, sep='_')   #组合出一个家庭变量
```

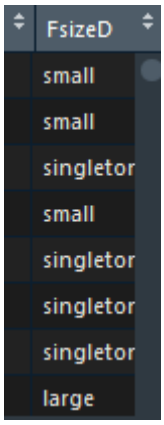
Surname	Fsize	Family
Braund	2	Braund_2
Cumings	2	Cumings_2
Heikkinen	1	Heikki
Futrelle	2	Futrelle_2
Allen	1	Allen_1
Moran	1	Moran_1
McCarthy	1	McCarthy_1
Palsson	5	Palsson_5

```
ggplot(full[1:891,], aes(x = Fsize, fill = factor(Survived))) +     #取 train 数据集
  geom_bar(stat='count', position='dodge') +    #dodge 分组并排，stat=count 是计数
  scale_x_continuous(breaks=c(1:11)) +          #x 轴连续的有 11 个值
  labs(x = 'Family Size') +                     #轴标签
```

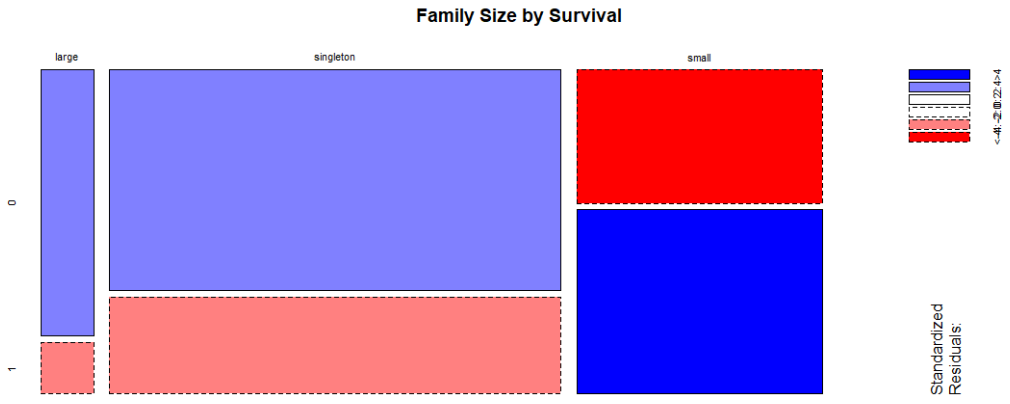
theme_few()



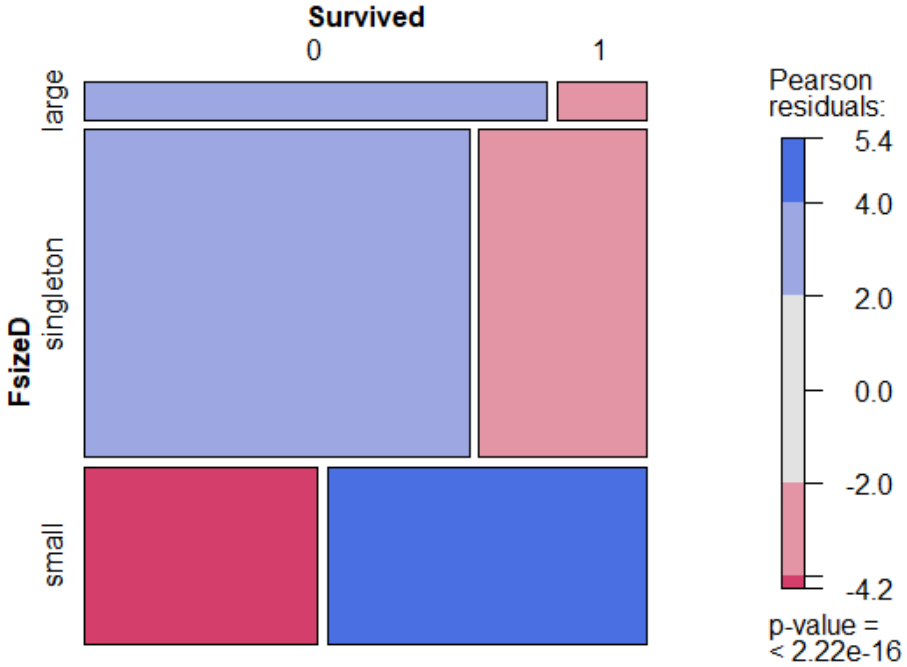
full$FsizeD[full$Fsize == 1] <- 'singleton'　　#1 个
full$FsizeD[full$Fsize < 5 & full$Fsize > 1] <- 'small'　　#2、3、4
full$FsizeD[full$Fsize > 4] <- 'large'　　#5。。。



mosaicplot(table(full$FsizeD, full$Survived),
　　main='Family Size by Survival', shade=TRUE)　　#马赛克图



library(vcd)
mosaic(~FsizeD+Survived, data=full, shade=TRUE, legend=TRUE)　　#第二种马赛图

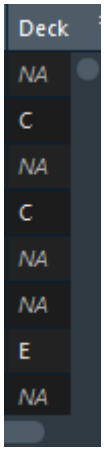


####

full$Cabin[1:28] #列举一下有很多缺失值

#strsplit(full$Cabin[2], NULL)[[1]] #例举 strsplit(full$Cabin[2], NULL)选出 cabin 的第二个 NULL 相当于“” 然后 strsplit 输出是一个 list 所以用[[1]]

full$Deck<-factor(sapply(full$Cabin, function(x) strsplit(x, NULL)[[1]][1])) #最后选取一列的一值

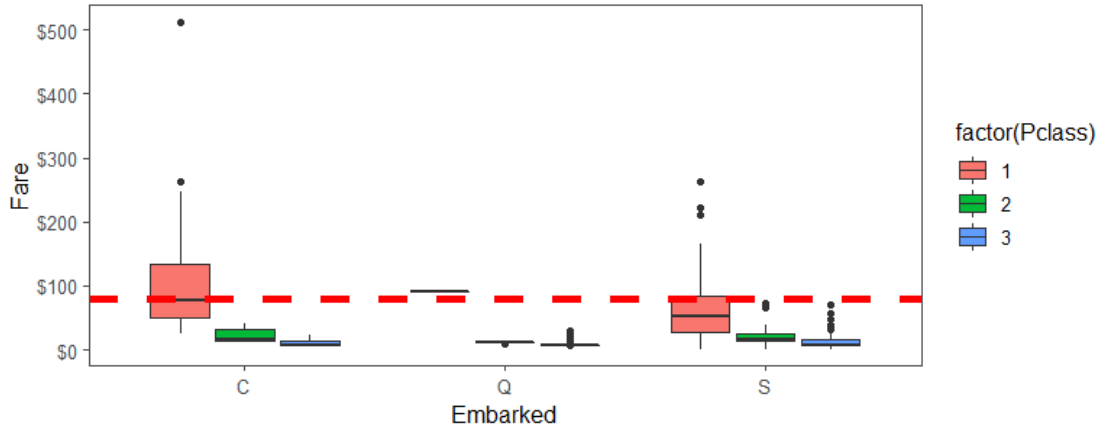


修复 Embarked 缺失值

full[c(62, 830), 'Embarked'] #62 和 830 行，embarked 列缺失

```
cat(paste('We will infer their values for
          **embarkment** based on present data that we can imagine may be relevant:
          **passenger class** and **fare**. We see that they paid<b> $',
          full[c(62, 830), 'Fare'][[1]][1], '</b>and<b> $',
          full[c(62, 830), 'Fare'][[1]][2], '</b>respectively and their classes are<b>',
```

```
            full[c(62, 830), 'Pclass'][[1]][1], '</b>and<b>',
            full[c(62, 830), 'Pclass'][[1]][2], '</b>. So from where did they embark?'))
embark_fare <- full %>%
  filter(PassengerId != 62 & PassengerId != 830)     #管道函数将 full 传递给 filter，选择出
                                                      PassengerId != 62 & PassengerId != 830
ggplot(embark_fare, aes(x = Embarked, y = Fare, fill = factor(Pclass))) +#
  geom_boxplot() +
  geom_hline(aes(yintercept=80),                                  #加辅助线，截距 80 处
             colour='red', linetype='dashed', lwd=2) +            #颜色，线形，线宽
  scale_y_continuous(labels=dollar_format()) +                    #增加了一个美元符号
  theme_few()
```

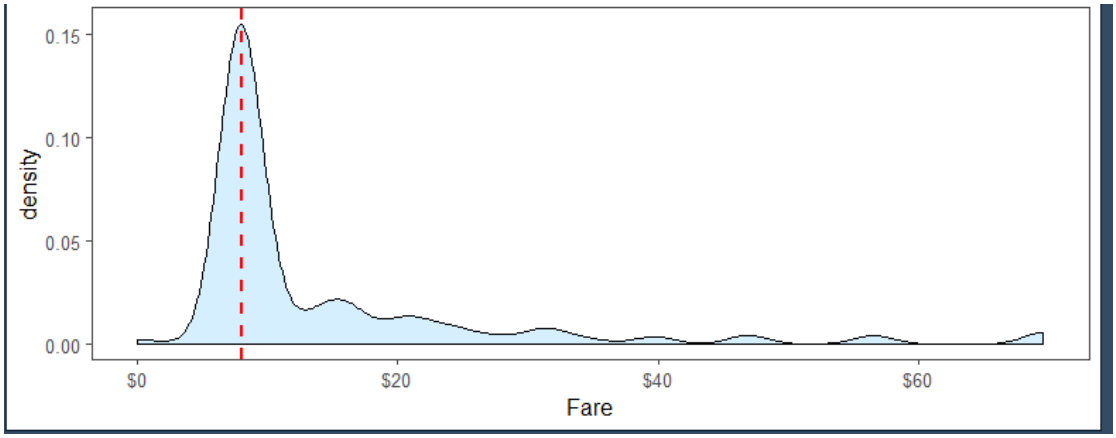



```
full$Embarked[c(62, 830)] <- 'C'                    #1 等仓 80 票价最可能在 c，所以给这
几个标记为 C
```

#修复 fare 的缺失值

```
full[1044, ]

ggplot(full[full$Pclass == '3' & full$Embarked == 'S', ],          #那些数据要展示，x 轴是 fare
       aes(x = Fare)) +
  geom_density(fill = '#99d6ff', alpha=0.4) +     #密度图颜色是#99d6ff，透明度 0.4
  geom_vline(aes(xintercept=median(Fare, na.rm=T)),       #画一条垂线——中位数线，去除
                                                          na 值
             colour='red', linetype='dashed', lwd=1) +    #特征
  scale_x_continuous(labels=dollar_format()) +     #增加一个美元符号
  theme_few()
```



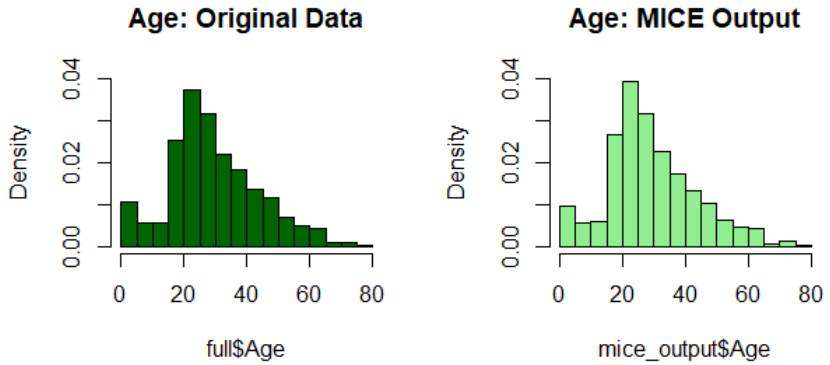

```
full$Fare[1044] <- median(full[full$Pclass == '3' & full$Embarked == 'S', ]$Fare, na.rm = TRUE)
#对 na 值进行替换
which(is.na(full$Fare))      #查看 na 值得下标
```

#预测，搞出完整的 age 数据

```
sum(is.na(full$Age))

factor_vars <- c('PassengerId','Pclass','Sex','Embarked',
                 'Title','Surname','Family','FsizeD')

full[factor_vars] <- lapply(full[factor_vars], function(x) as.factor(x))
set.seed(129)
mice_mod <- mice(full[, !names(full) %in%
c('PassengerId','Name','Ticket','Cabin','Family','Surname','Survived')], method='rf')
##把非 c（）中的选出来，rf 是随机森林
mice_output <- complete(mice_mod) #输出完整的数据
par(mfrow=c(1,2))
hist(full$Age, freq=F, main='Age: Original Data',
     col='darkgreen', ylim=c(0,0.04))
hist(mice_output$Age, freq=F, main='Age: MICE Output',
     col='lightgreen', ylim=c(0,0.04))
```



二者一样

```
full$Age <- mice_output$Age
```

```
sum(is.na(full$Age))
```

错

```
ggplot(full[1:891,], aes(Age, fill = factor(Survived))) +
  geom_histogram() +
  facet_grid(.~Sex) +     #每个 sex 独立图，变成一个单行
  theme_few()
full$Child[full$Age < 18] <- 'Child'
full$Child[full$Age >= 18] <- 'Adult'
table(full$Child, full$Survived)
```

```
> table(full$Child, full$Survived)

        0   1
Adult 372 229
Child  52  61
```

```
full$Mother <- 'Not Mother'
full$Mother[full$Sex == 'female' & full$Parch > 0 & full$Age > 18 & full$Title != 'Miss'] <-
'Mother'
table(full$Mother, full$Survived)
```

```
> table(full$Mother, full$Survived)

               0   1
Mother        15  37
Not Mother   534 305
>
```

```
full$Child   <- factor(full$Child)
full$Mother <- factor(full$Mother)
md.pattern(full)


train <- full[1:891,]
test <- full[892:1309,]

set.seed(754)

# Build the model (note: not all possible variables are used)
rf_model <- randomForest(factor(Survived) ~ Pclass + Sex + Age + SibSp + Parch +
                                            Fare + Embarked + Title +
                                            FsizeD + Child + Mother,
                                          data = train)

# Show model error
plot(rf_model, ylim=c(0,0.36))
legend('topright', colnames(rf_model$err.rate), col=1:3, fill=1:3)
```

```
# Get importance
importance       <- importance(rf_model)
varImportance <- data.frame(Variables = row.names(importance),
                                        Importance = round(importance[ ,'MeanDecreaseGini'],2))

# Create a rank variable based on importance
rankImportance <- varImportance %>%
   mutate(Rank = paste0('#',dense_rank(desc(Importance))))

# Use ggplot2 to visualize the relative importance of variables
ggplot(rankImportance, aes(x = reorder(Variables, Importance),
                                      y = Importance, fill = Importance)) +
   geom_bar(stat='identity') +
   geom_text(aes(x = Variables, y = 0.5, label = Rank),
                hjust=0, vjust=0.55, size = 4, colour = 'red') +
   labs(x = 'Variables') +
   coord_flip() +
   theme_few()
```