数据分析与处理技术

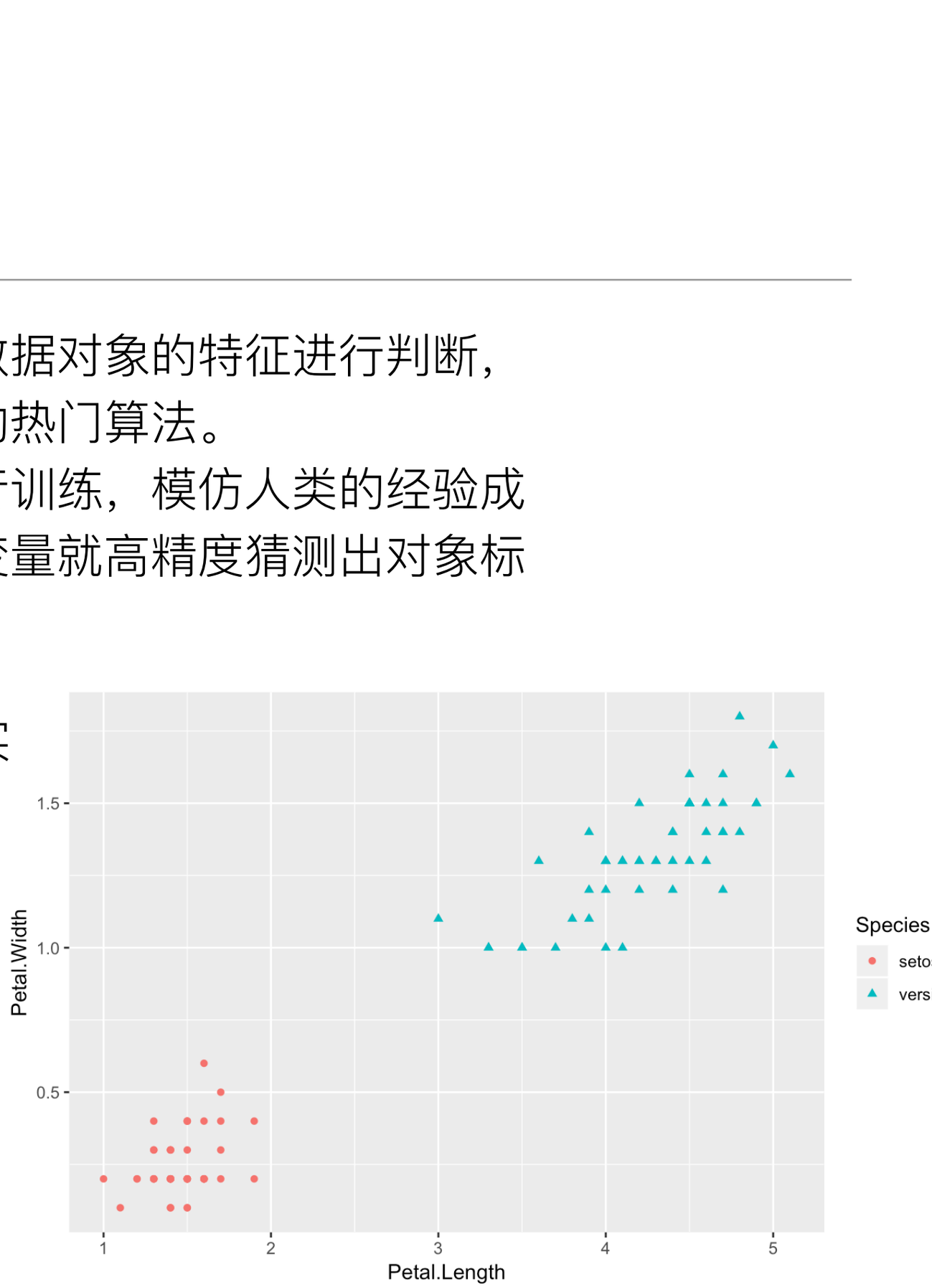
分类预测

线性分类模型

分类问题是根据自变量观察到的数据对数据对象的特征进行判断，
这种方法是当前数据分析和数据挖掘中的热门算法。
利用已经有类别标签的数据集对模型进行训练，模仿人类的经验成长过程，逐步实现模型能够通过读入自变量就高精度猜测出对象标签特征。

线性分类器是分类算法里最基础也是最实用的方法，从iris划分类别开始。

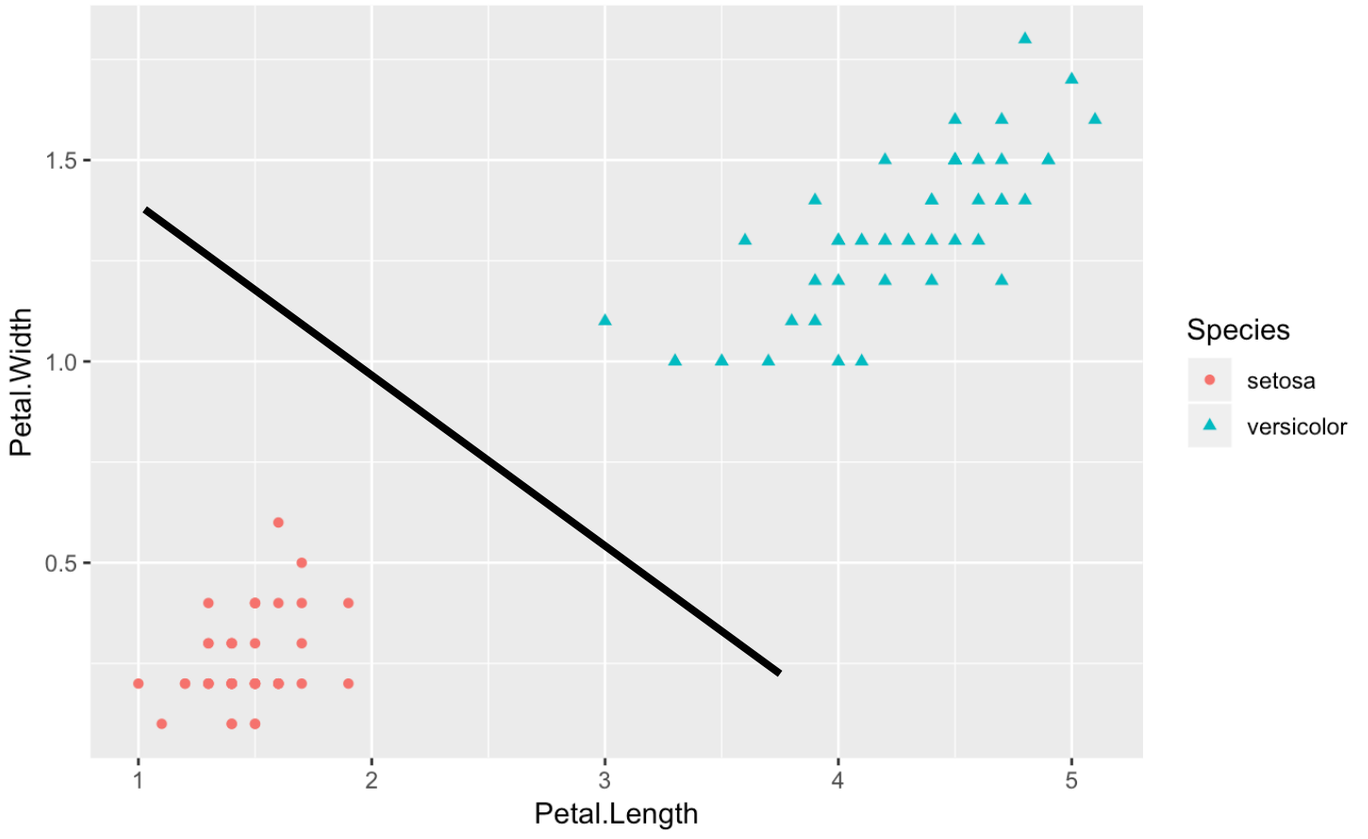
右侧数据仅取了iris后三列变量，并且Species中仅保留了’versicolor’和’setosa’两个属性值对象



加载必要工具包，lda函数属于MASS包

```
library(MASS)
library(ggplot2)
```

lda相当于在右侧散点图中画出一条清晰的分类界限，将两类数据点隔开，那么当再有新数据时我们将直接根据新数据在分类线哪侧判断数据所属类别



上图分类线为手工画线

画出待分类数据散点图

```
iris2=iris[-which(iris$Species=='virginica'),3:5]
iris2$Species=factor(iris2$Species)
ggplot(data=iris2)+
  geom_point(aes(Petal.Length,Petal.Width,color=Species,shape=Species))
```

线性分类器的功能是找出一个标准，在数据的分布中将类型划分开，为了进行验证，将数据集分成两组train和test分别用于训练充当分类器的模型的数据和用于检验的数据

```
s=sample(1:nrow(iris2),nrow(iris2)*0.7)
train=iris2[s,]
test=iris2[-s,]
```

lda函数形式上非常类似线性回归的lm函数，函数的结果通常称为分类器

```
classifier=lda(Species~Petal.Length+Petal.Width,data=train)
```

```
y_pred=predict(classifier,type='response',newdata = test)
```

利用分类器对test数据进行预测

进一步，看模型对测试数据集test分类是否正确

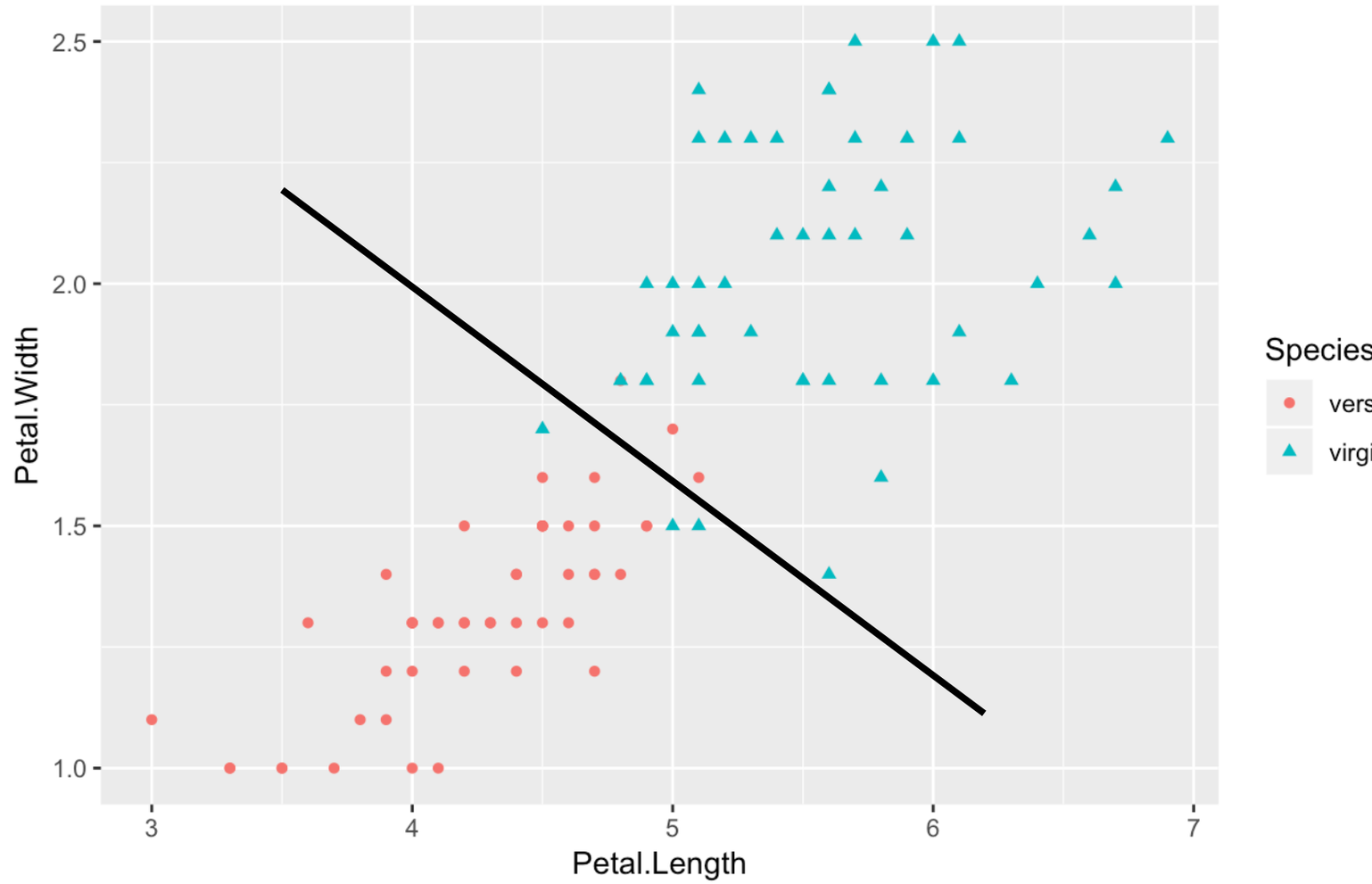
```
table(test$Species,y_pred$class)
```

```
            setosa versicolor
setosa          15          0
versicolor       0         15
```

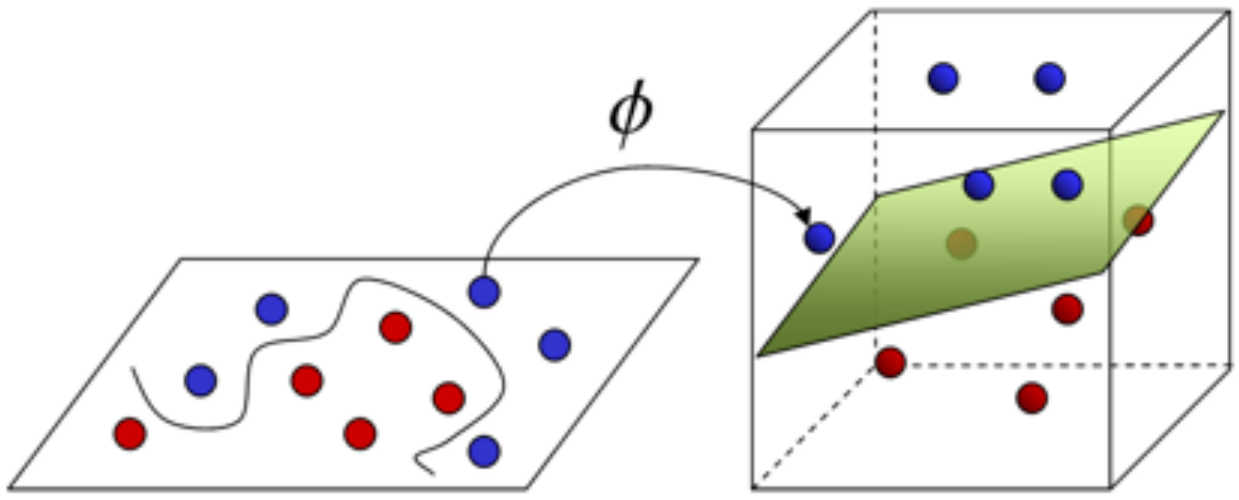
线性可分的情况毕竟是少数，更多的数据无法完全用一条直接绝对准确分开两类。为了找到最合适的分类直线位置，需要创建一个标准，即误分类损失

$$L(a,b)=-\sum_{x_i\in M} y_i(ax_i+b)$$

那么剩下任务就是找到使 $L(a,b)$ 能够最小化的a和b



对与更为复杂的分类问题，有时线性分类器无论如何都无法将散乱的数据点合理分开。
或许是我们观察事物的维度和角度不够，信息不足以支撑起分类模型，那么可以通过映射到高维空间的方法找到足够的分类空间。
支持向量机(Support Vector Machine)通过kernel函数实现了这种分类方法，如下图所示。



以iris为例，这次让四个数值型变量都参与模型训练，采用e1071包中的SVM函数实现支持向量机分类

```
s=sample(1:nrow(iris),nrow(iris)*0.7)
train1=iris[s,]
test1=iris[-s,]
```

对数据进行训练集和测试集拆分

```
classifier1=svm(Species~.,data=train1,type='C-classification',kernel='linear')
y_pred1=predict(classifier1,newdata = test1)
```

训练分类器，并预测

```
table(test1$Species,y_pred1)
           y_pred1
            setosa versicolor virginica
setosa          12          0         0
versicolor       0         14         1
virginica        0          0        18
```

检查分类效果

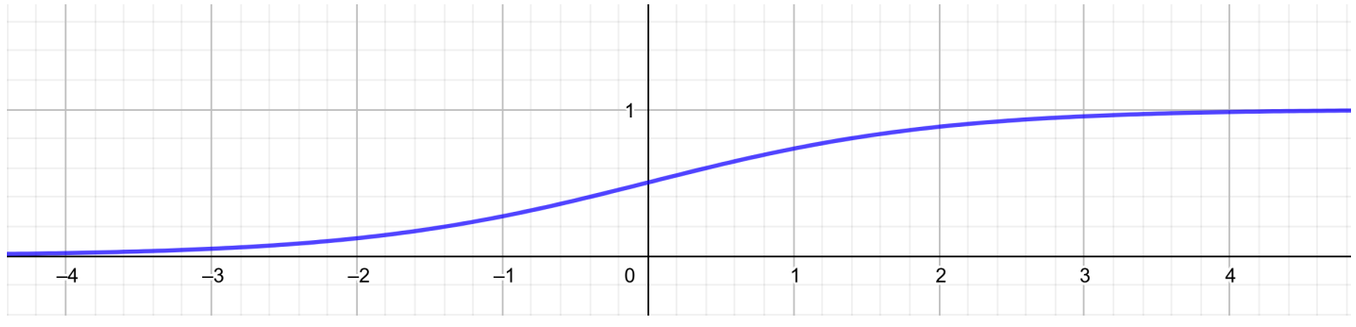
二元选择模型-Logistic模型

二元选择模型的y只有两个分类取值，即定义为1和0

logistic回归拟合事件发生的概率

$$P(y_i = 1 \mid X_i) = \frac{e^{\beta X_i}}{1 + e^{\beta X_i}}$$

$$P(y_i = 0 \mid x_i) = 1 - \frac{e^{\beta X_i}}{1 + e^{\beta X_i}}$$



其中X代表了一个含参数变量线性组合 $\beta X_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n$

那么，发生了为1的事件和它的概率相乘： $1 \cdot P(y_i = 1 \mid X_i)$

发生了为0的事件和它的概率相乘： $0 \cdot P(y_i = 0 \mid X_i)$

所有这些加一起就是似然函数 $L = \sum 1 \cdot P(y_i = 1 \mid X_i) + \sum 0 \cdot P(y_i = 0 \mid X_i)$

当然我们要找到能够令L最大化的参数组合 $\beta_0, \beta_1, \beta_2 \cdots \beta_n$

案例：Titanic数据集中挖掘生存预测信息

将前变作业里涉及的Titanic数据集作为案例，当分析到最后一步时使用二元分类模型来分析和预测不同身份特征人遇难的概率

将891行原训练数据集和剩下的418行数据拼接组成完整Titanic数据集

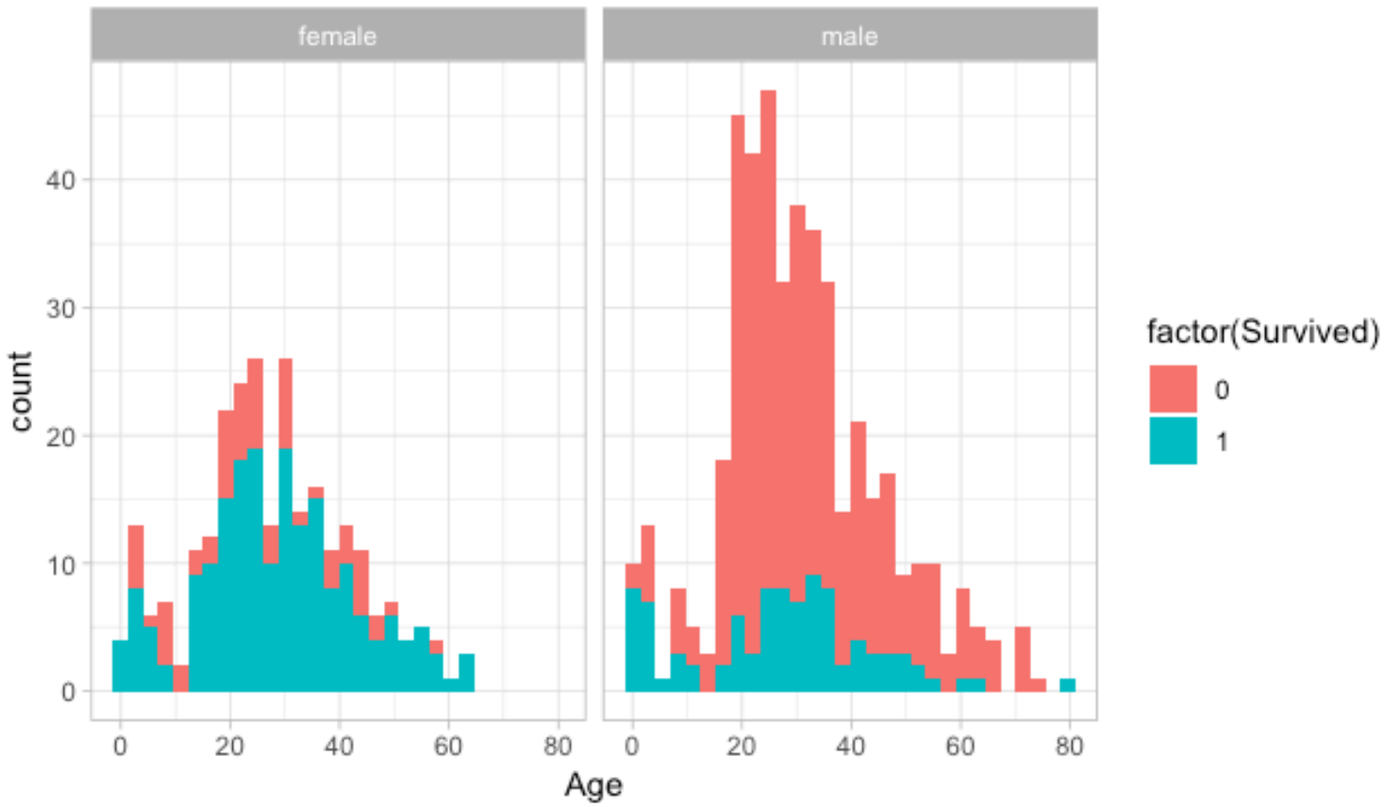
首先将Titanictest和存放后418行Survived变量的两个数据集合并，再与原891行数据拼接在一起组成完整数据集

```
newtest=merge(Titanictest,gender_submission)
Titanic=bind_rows(Titanic1,newtest)
```

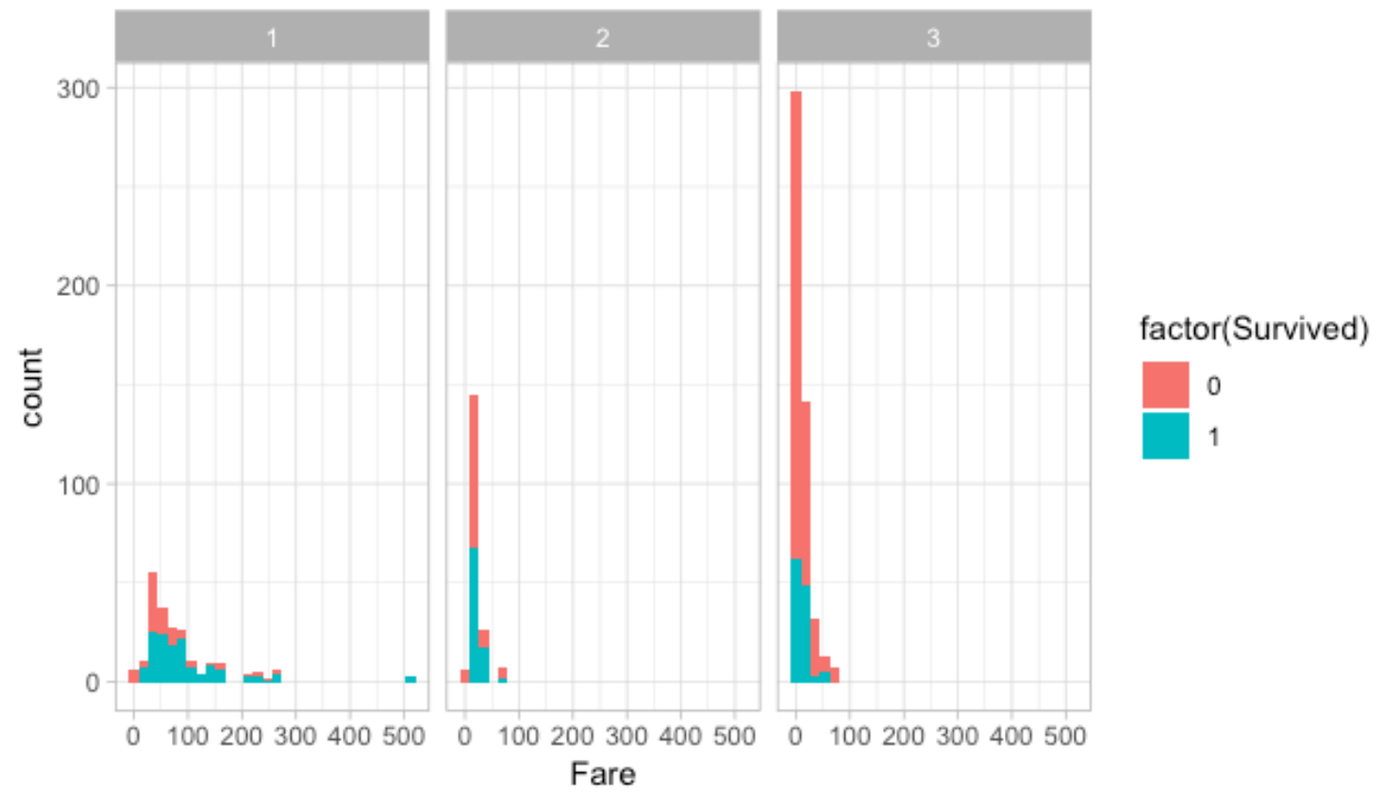
直观的探索性分析

哪些信息与生存相关

```
ggplot(data=Titanic)+
    geom_histogram(aes(x=Age,fill=factor(Survived)))+
    facet_grid(.~Sex)+theme_light()
```




```
ggplot(data=Titanic)+
    geom_histogram(aes(x=Fare,fill=factor(Survived)))+
    facet_grid(.~Pclass)+theme_light()
```



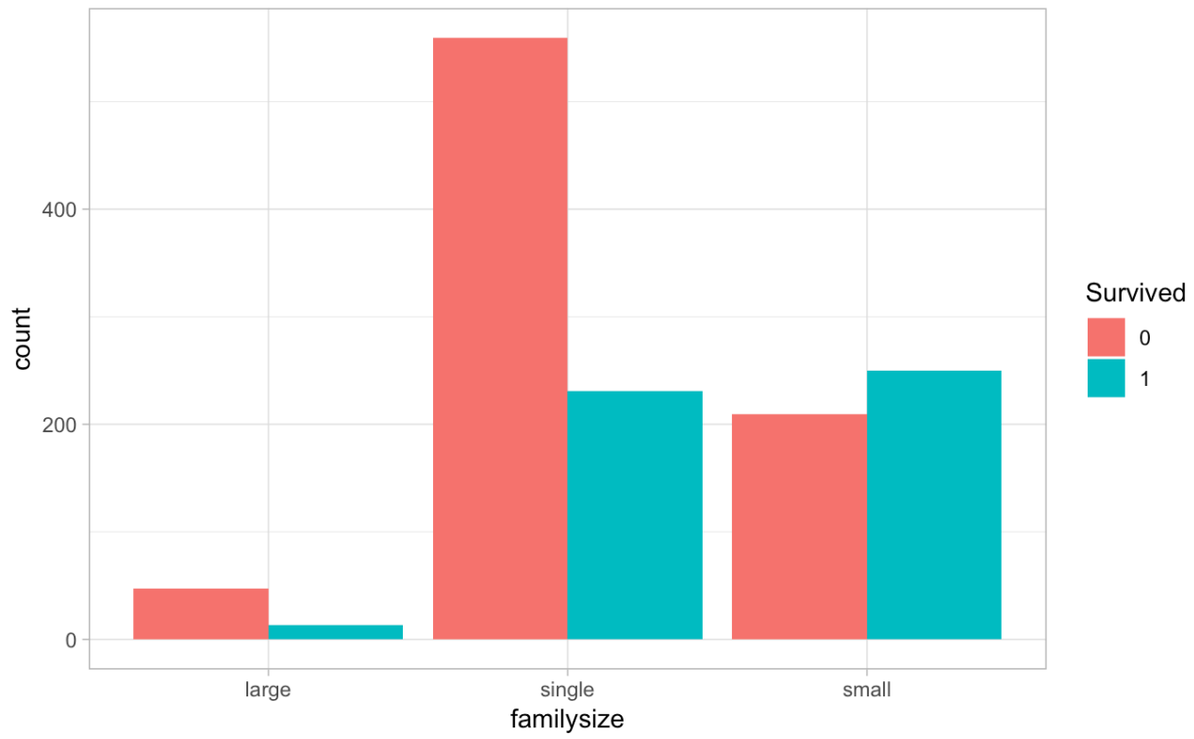
```
fsize=Titanic$SibSp+Titanic$Parch #将兄弟姐妹数和家庭子女数加起来
```

```
> a=ifelse(fsize==0,'single',
+          ifelse(fsize>4,'large','small'))
> Titanic$familysize=factor(a)
> table(Titanic$familysize)

 large single  small
    60    790    459
```

不难想象，灾难发生时或许家庭背景会对人产生不同影响，分析每个人家庭背景，即家庭成员规模大小。将数据集中兄弟姐妹数量和直系子女数量组合起来代表家庭成员规模

```
ggplot(Titanic)+
  geom_bar(aes(x=familysize,y=..count..,fill=Survived),position='dodge')+
  theme_light()
```



使用logistic模型分析Titanic数据集

首先拆成训练集和测试集两个数据集，并初步筛选变量

```
train=Titanic[1:891,c('Survived','Pclass','Sex','familysize')]
test=Titanic[892:1309,c('Survived','Pclass','Sex','familysize')]
```

利用glm函数训练模型分类器

```
classifier = glm(Survived ~ ., family = binomial(link='logit'), data = train)
summary(classifier)
```

检验模型判断对错情况，当然阈值按照通常习惯采用0.5

```
prob_pred=predict(classifier,type = 'response',newdata = test)
y_pred=ifelse(prob_pred>0.5,1,0)
table(test$Survived,y_pred>0.5)
```

```
   FALSE TRUE
0   266    0
1     4  148
```

检验分类效果

计算分类器在测试集上的具体精确程度

```
> e=sum(test$Survived==y_pred)/nrow(test)
> paste('Accuracy:',round(e,4))
[1] "Accuracy: 0.9904"
```

更为详细的，做出ROC图检验分类器能力

```
fitpred=prediction(prob_pred,test$Survived)
fitperf=performance(fitpred,'tpr','fpr')
plot(fitperf,col='red',main='ROC Curve')
abline(a=0,b=1,lwd=2,lty=2,col='grey')
```