Credit Card Default

Aymen Rumi

5/8/2020

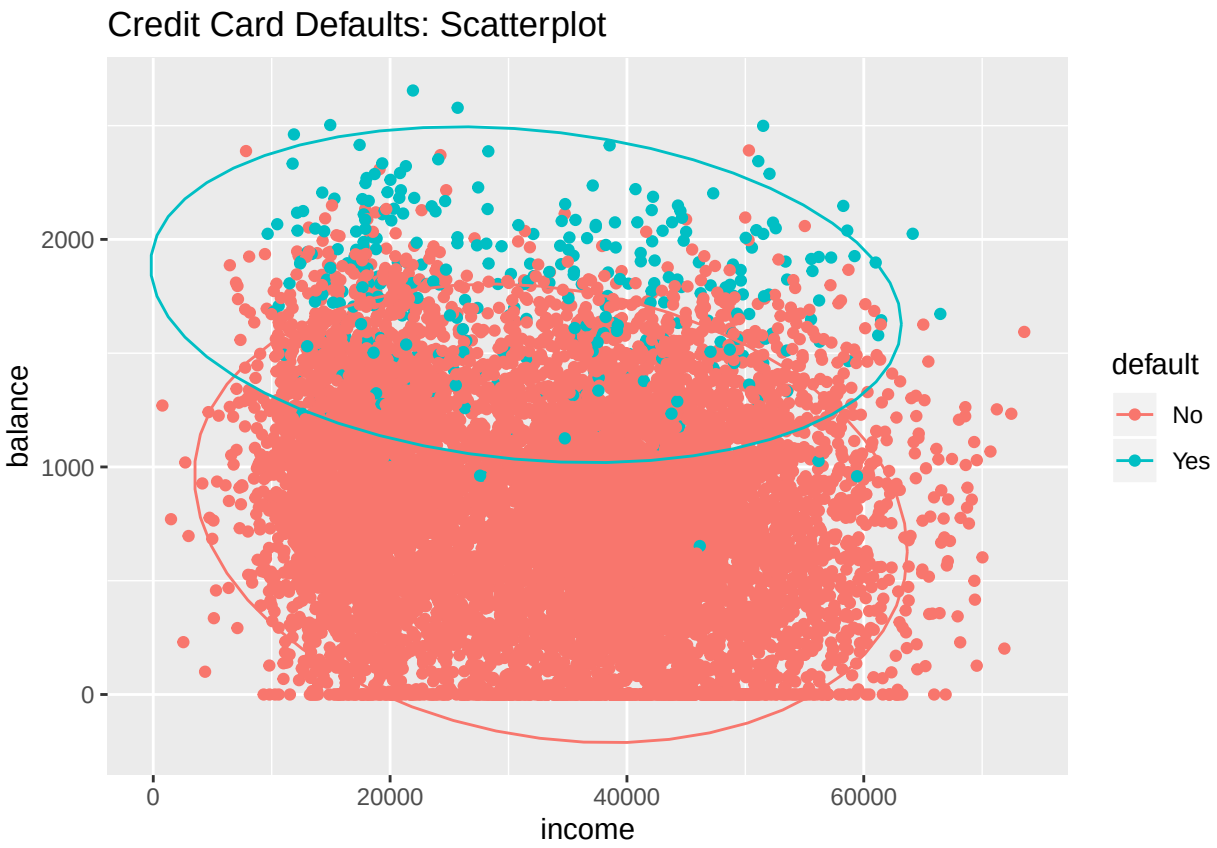
Overview

We will analyze and build a model for Credit Card Default data, with response being binary variable "default" and predictors being binary variable "student" and continuous variables balance & income

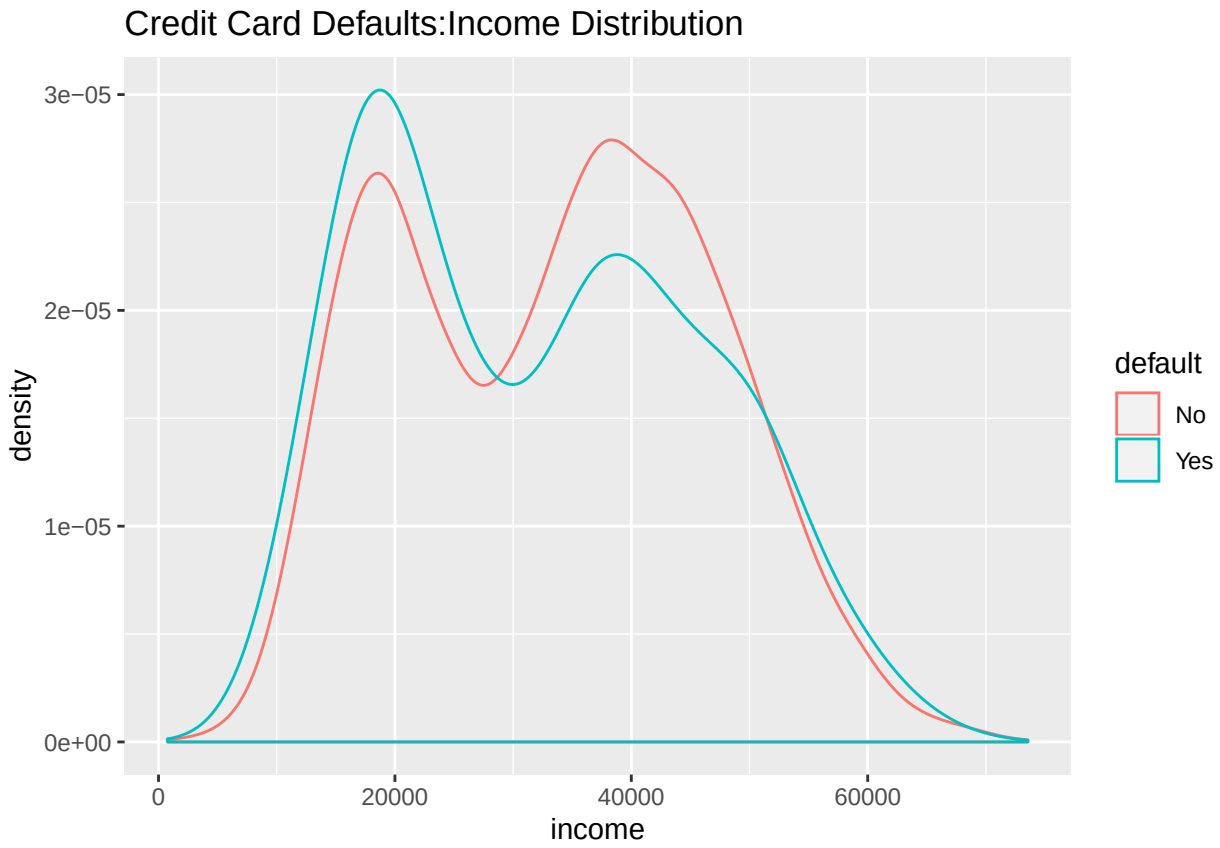
Plotting & Visualization

We explore the relationships between our repsonse and predictors with visualizations

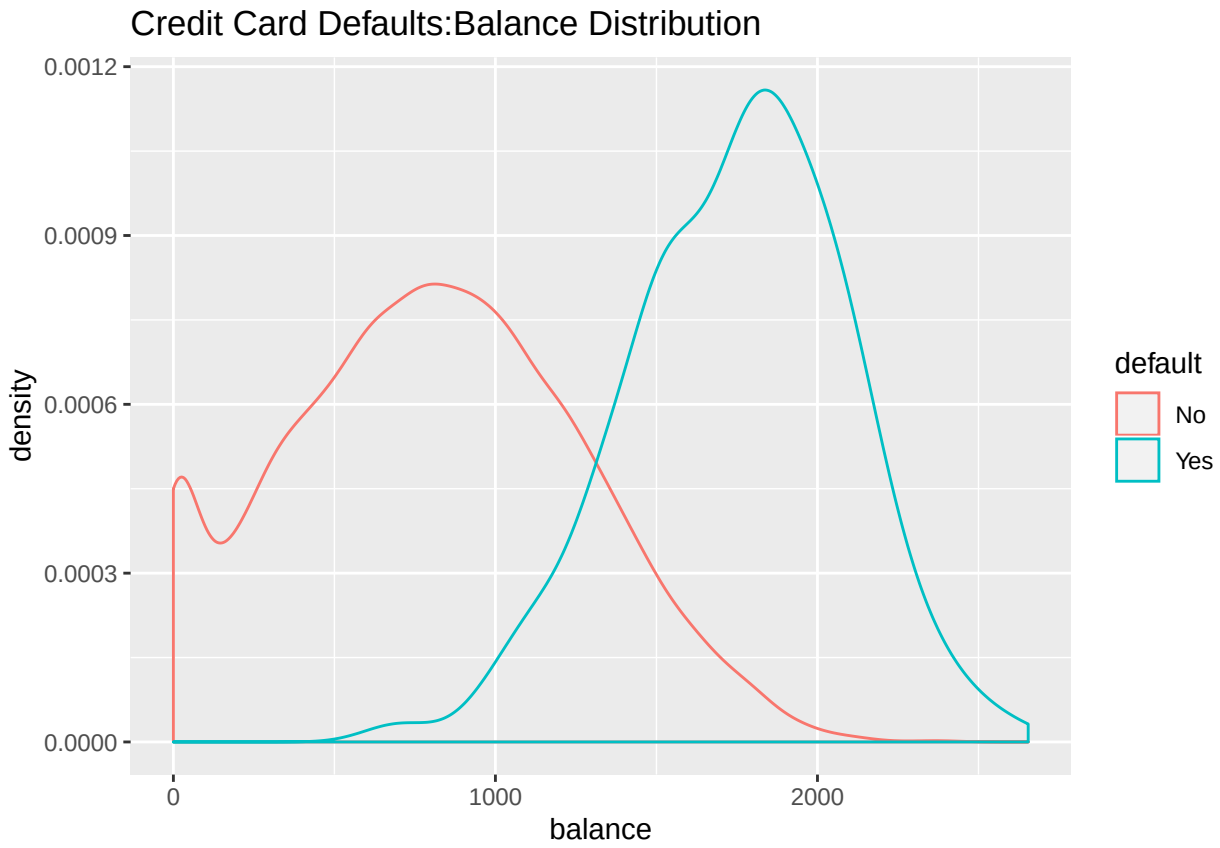
```
ggplot(data=Default,aes(x=income,y=balance,color=default))+geom_point()+stat_ellipse()+ggtitle("Credit
```



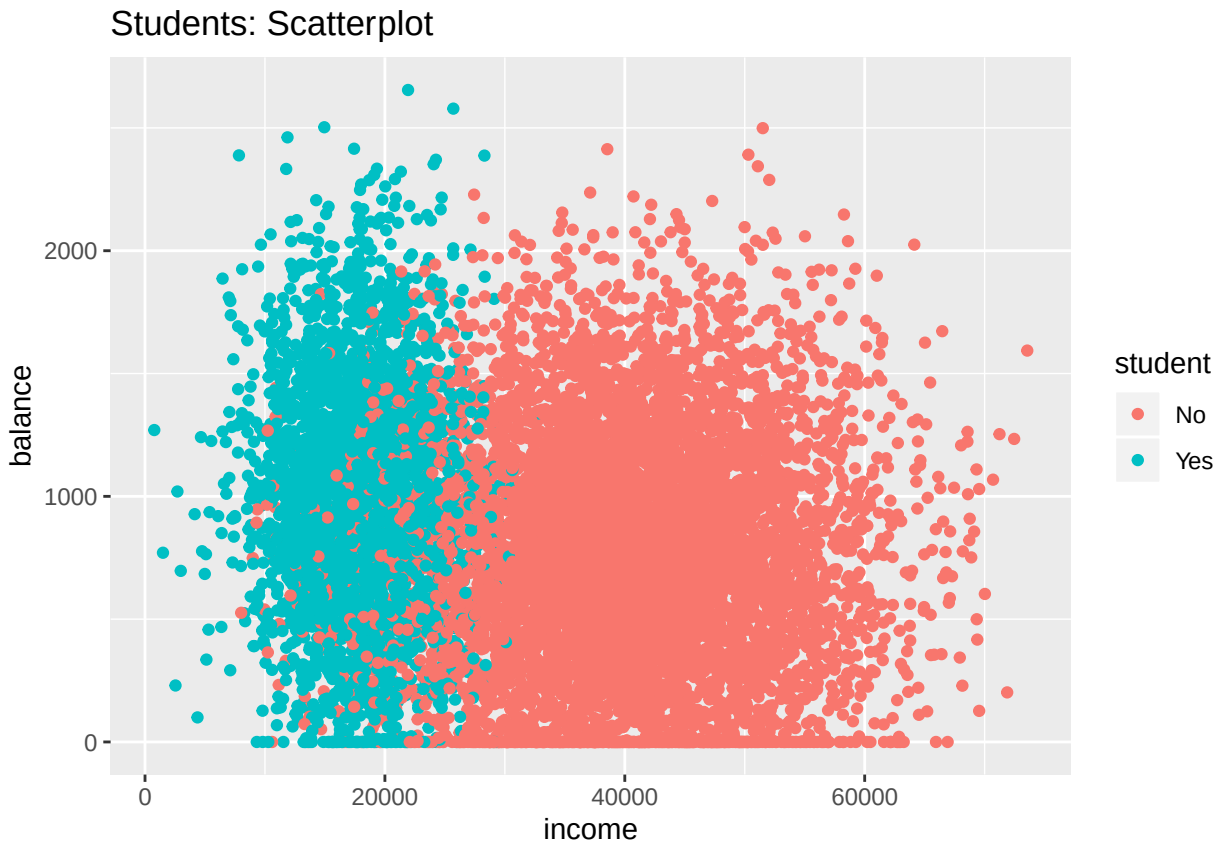

```
ggplot(data=Default,aes(x=income,color=default))+geom_density()+ggtitle("Credit Card Defaults:Income Di
```



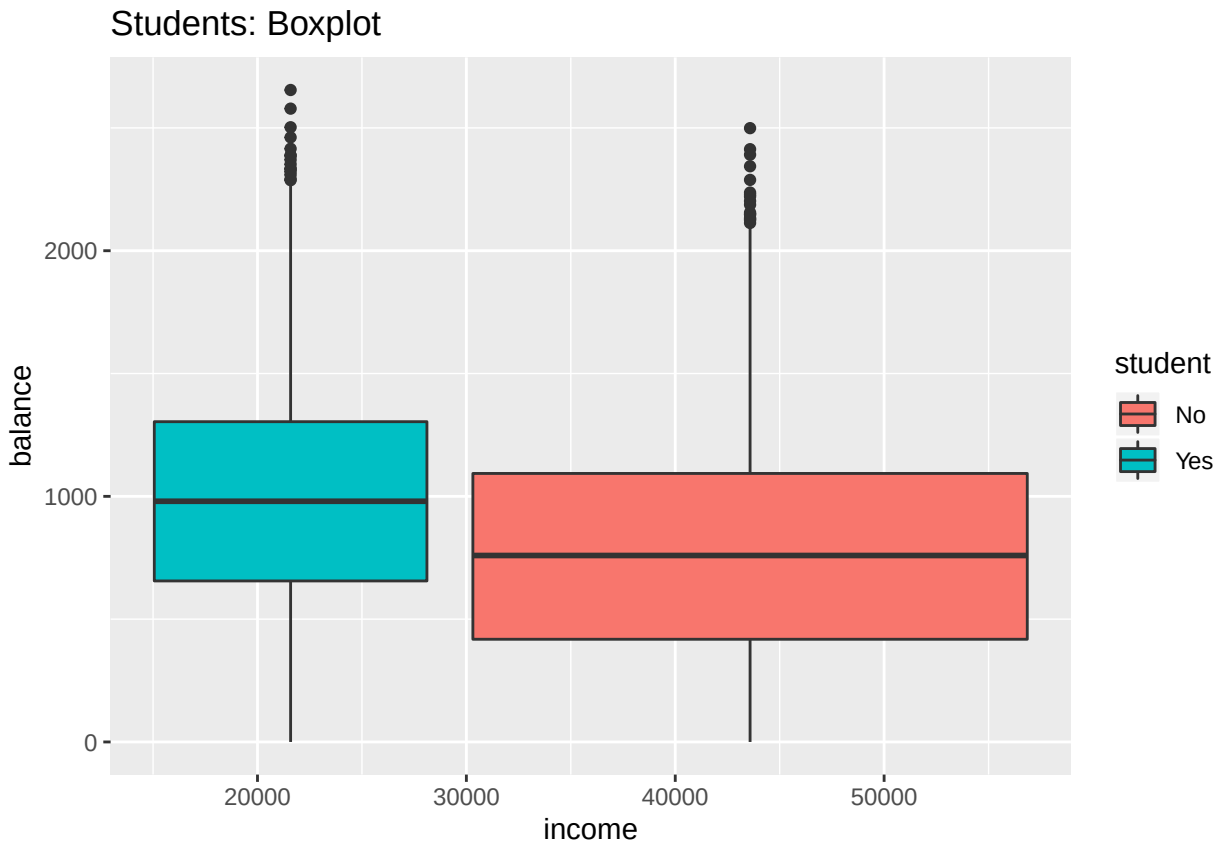

```
ggplot(data=Default,aes(x=balance,color=default))+geom_density()+ggtitle("Credit Card Defaults:Balance I
```



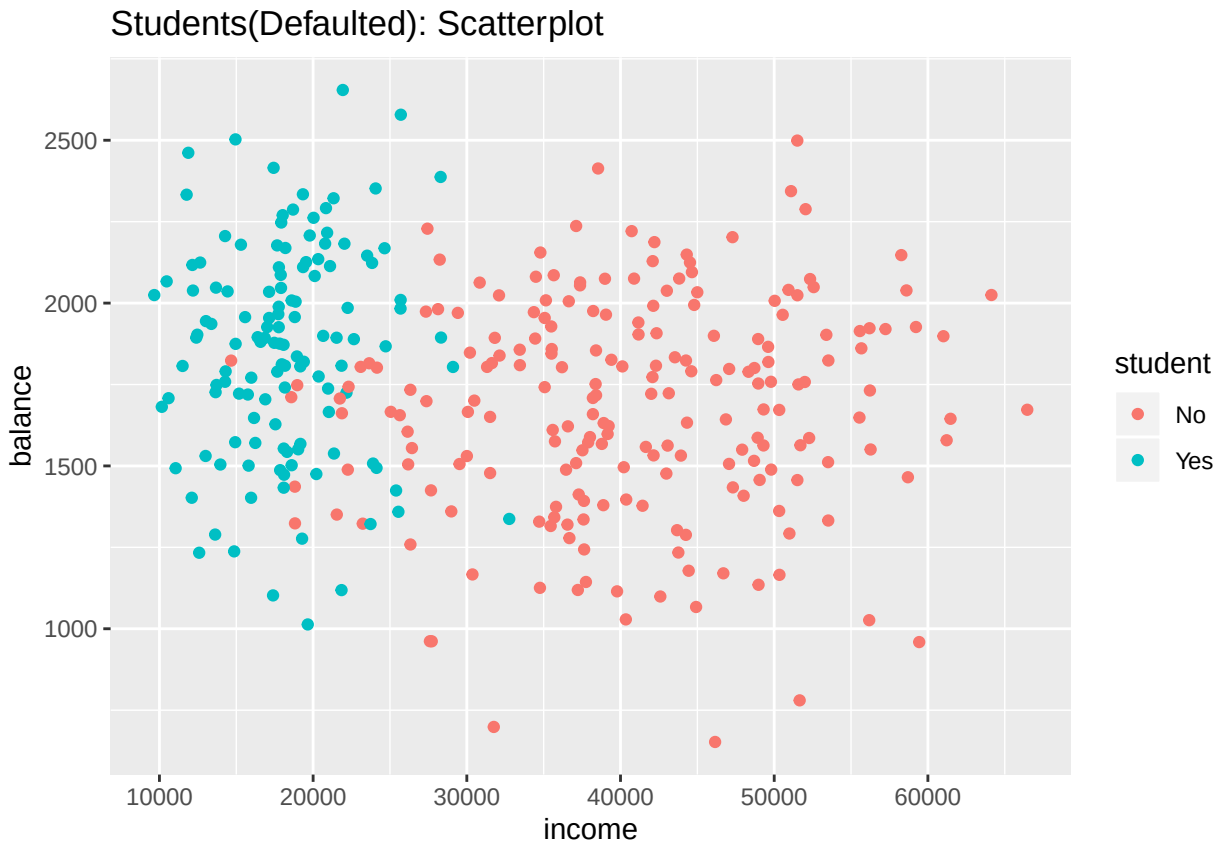

```
ggplot(data=Default,aes(x=income,y=balance,color=student))+geom_point()+ggtitle("Students: Scatterplot"
```



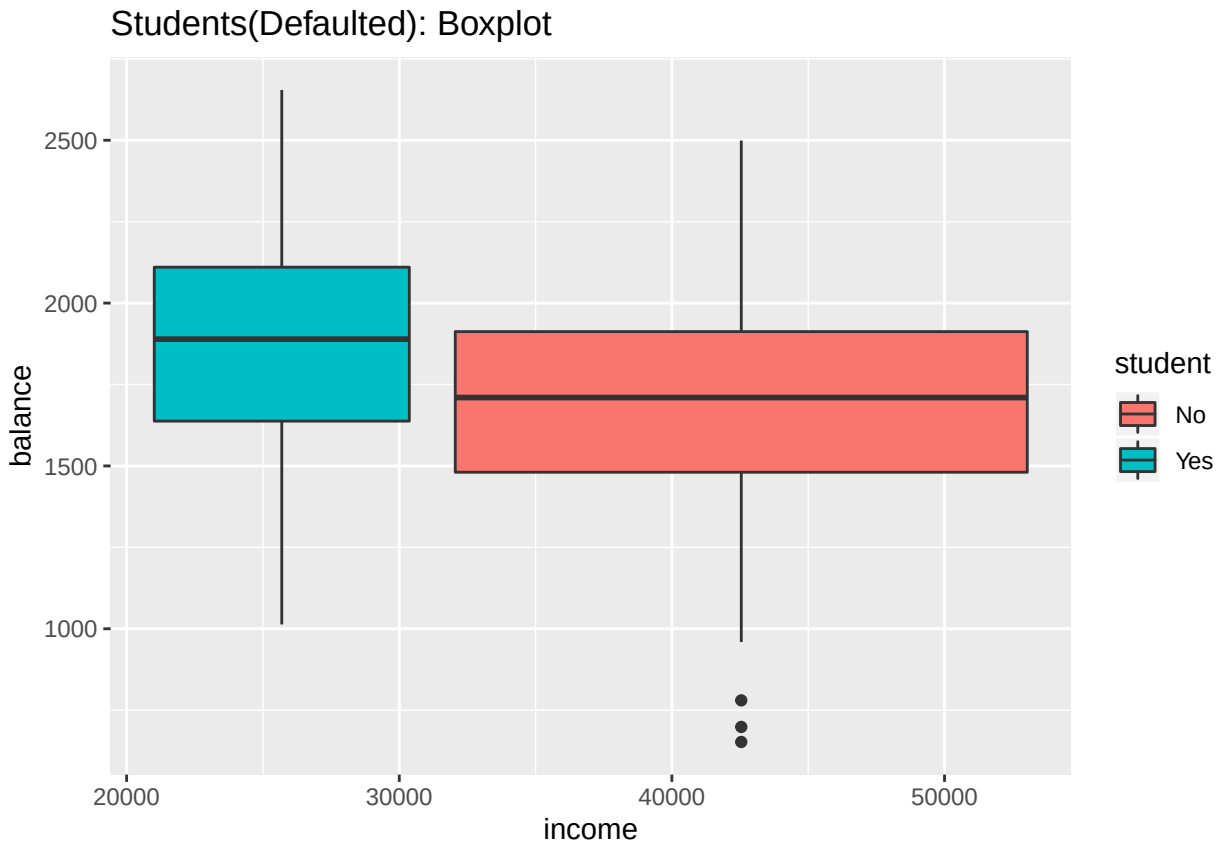

```
ggplot(data=Default, aes(x=income,y=balance, fill=student)) +ggtitle("Students: Boxplot")+
  geom_boxplot()
```



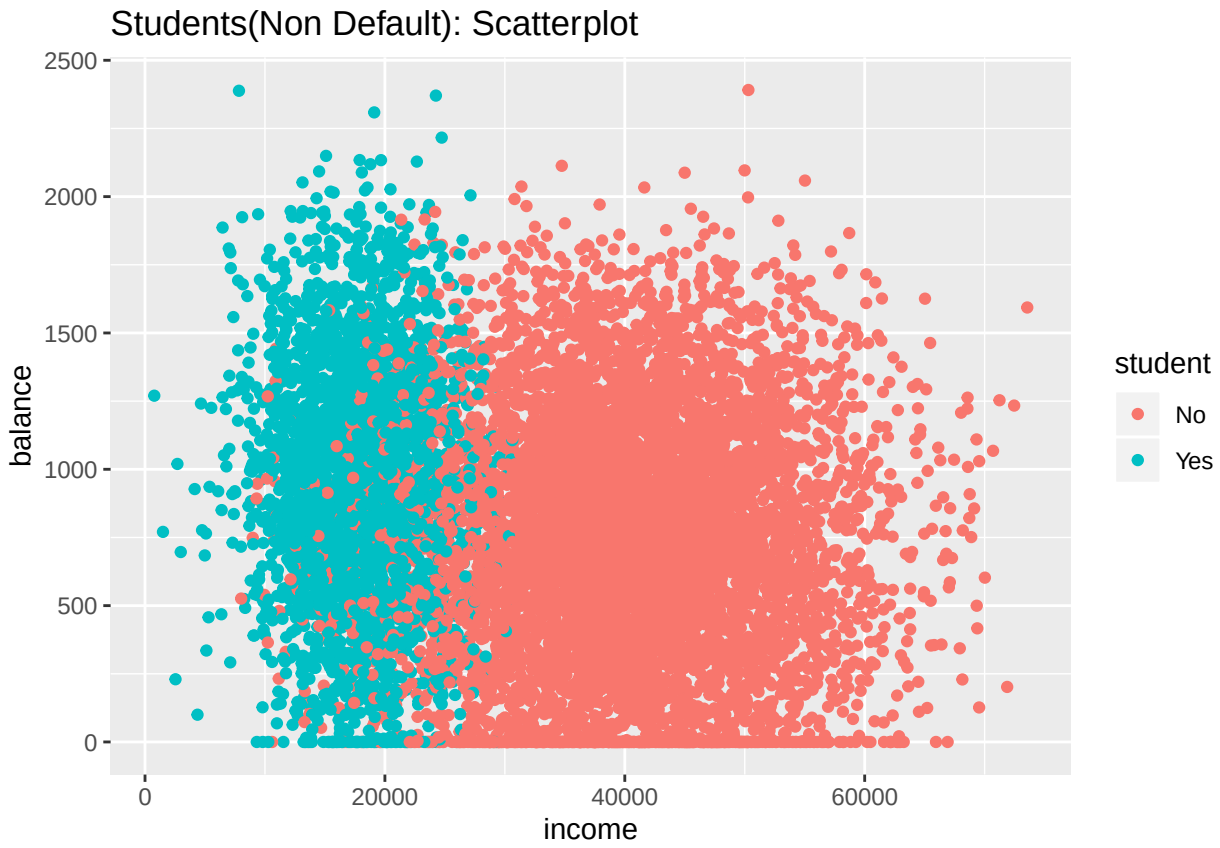

```
ggplot(data=subset(Default,default=="Yes"),aes(x=income,y=balance,color=student))+geom_point()+ggtitle(
```



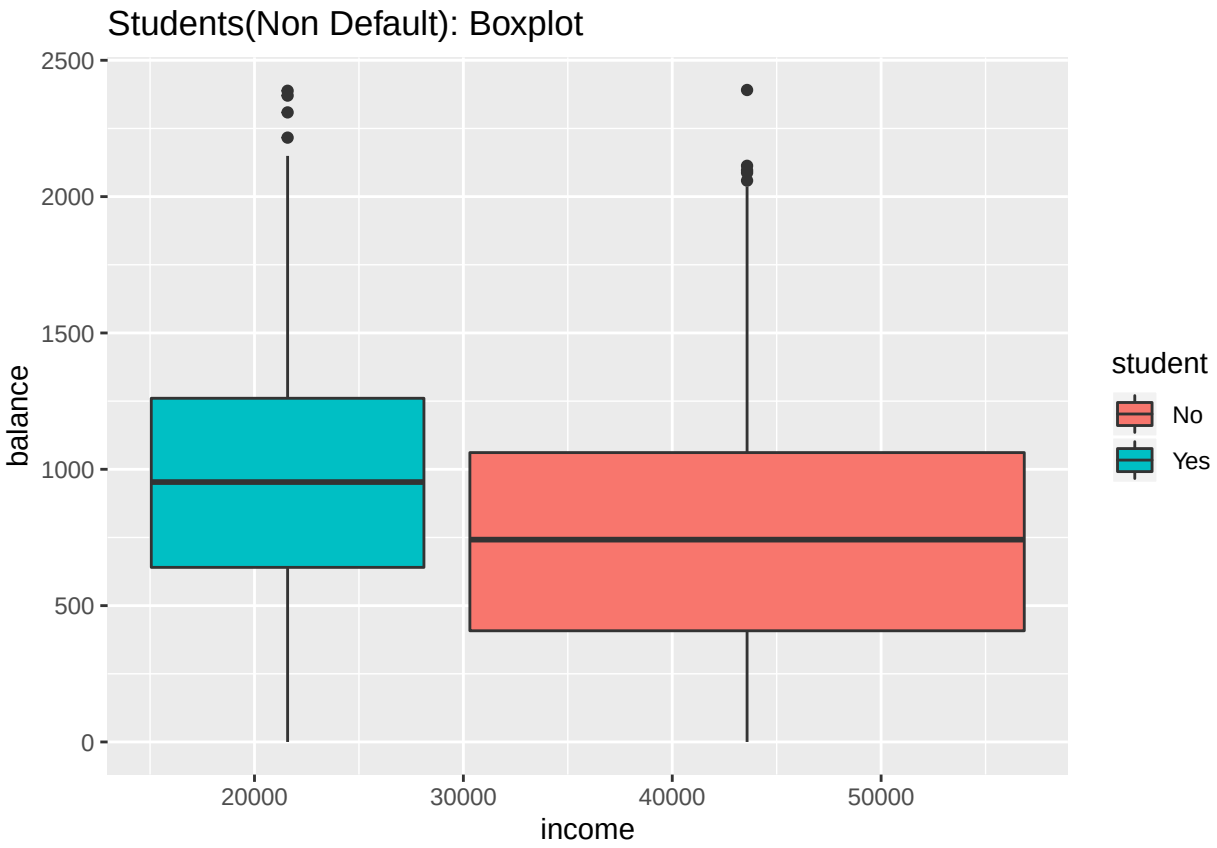

```
ggplot(data=subset(Default,default=="Yes"), aes(x=income,y=balance, fill=student)) +
  geom_boxplot()+ggtitle("Students(Defaulted): Boxplot")
```




```
ggplot(data=subset(Default,default=="No"),aes(x=income,y=balance,color=student))+geom_point()+ggtitle("S
```




```
ggplot(data=subset(Default,default=="No"), aes(x=income,y=balance, fill=student)) +
  geom_boxplot()+ggtitle("Students(Non Default): Boxplot")
```



Logistic Regression Model Training

```
attach(Default)
set.seed(1)

glm.fit=glm(default~balance+income,family=binomial,data=Default)

glm.probs=predict(glm.fit,type="response")

glm.pred=rep("No",length(Default$default))
glm.pred[glm.probs>0.5]="Yes"

table(glm.pred,default)
```

```
##         default
## glm.pred   No  Yes
##      No  9629  225
##      Yes   38  108
```

```
mean(glm.pred==default)
```

```
## [1] 0.9737
```

```
mean(glm.pred!=default)
```

```
## [1] 0.0263
```

K-Fold Cross Validation

```
cv.err=cv.glm(Default,glm.fit,K=5)
cv.err$delta[1]
```

```
## [1] 0.02149045
```

```
1-cv.err$delta[1]
```

```
## [1] 0.9785095
```

```
cv.err=cv.glm(Default,glm.fit,K=10)
cv.err$delta[1]
```

```
## [1] 0.02147942
```

```
1-cv.err$delta[1]
```

```
## [1] 0.9785206
```

```
cv.err=cv.glm(Default,glm.fit,K=100)
cv.err$delta[1]
```

```
## [1] 0.0214586
```

```
1-cv.err$delta[1]
```

```
## [1] 0.9785414
```

Bootstrap

```
classification_estimate<-function(data,index)
{
  response<-data$default[index]
  x1<-data$balance[index]
  x2<-data$income[index]

  glm.fit=glm(response~x1+x2,famil=binomial)

  glm.probs=predict(glm.fit,type="response")
```

```
  glm.pred=rep("No",length(Default$default))
  glm.pred[glm.probs>0.5]="Yes"

  table(glm.pred,default)

  return (mean(glm.pred==default))

}


boot(Default,statistic = classification_estimate,R=1000)
```

```
##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = Default, statistic = classification_estimate, R = 1000)
##
##
## Bootstrap Statistics :
##     original     bias    std. error
## t1*   0.9737 -0.0205257 0.001481616
```