Week4_TA_Session

Ajinkya Panaskar

2/1/2020

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 3.6.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.2
```

```
##
## 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(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.2
```

```
library(ISLR)
```

```
## Warning: package 'ISLR' was built under R version 3.6.2
```

```
library(ROCR)
```

```
## Warning: package 'ROCR' was built under R version 3.6.2
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'
## The following object is masked from 'package:stats':
##
##     lowess
```

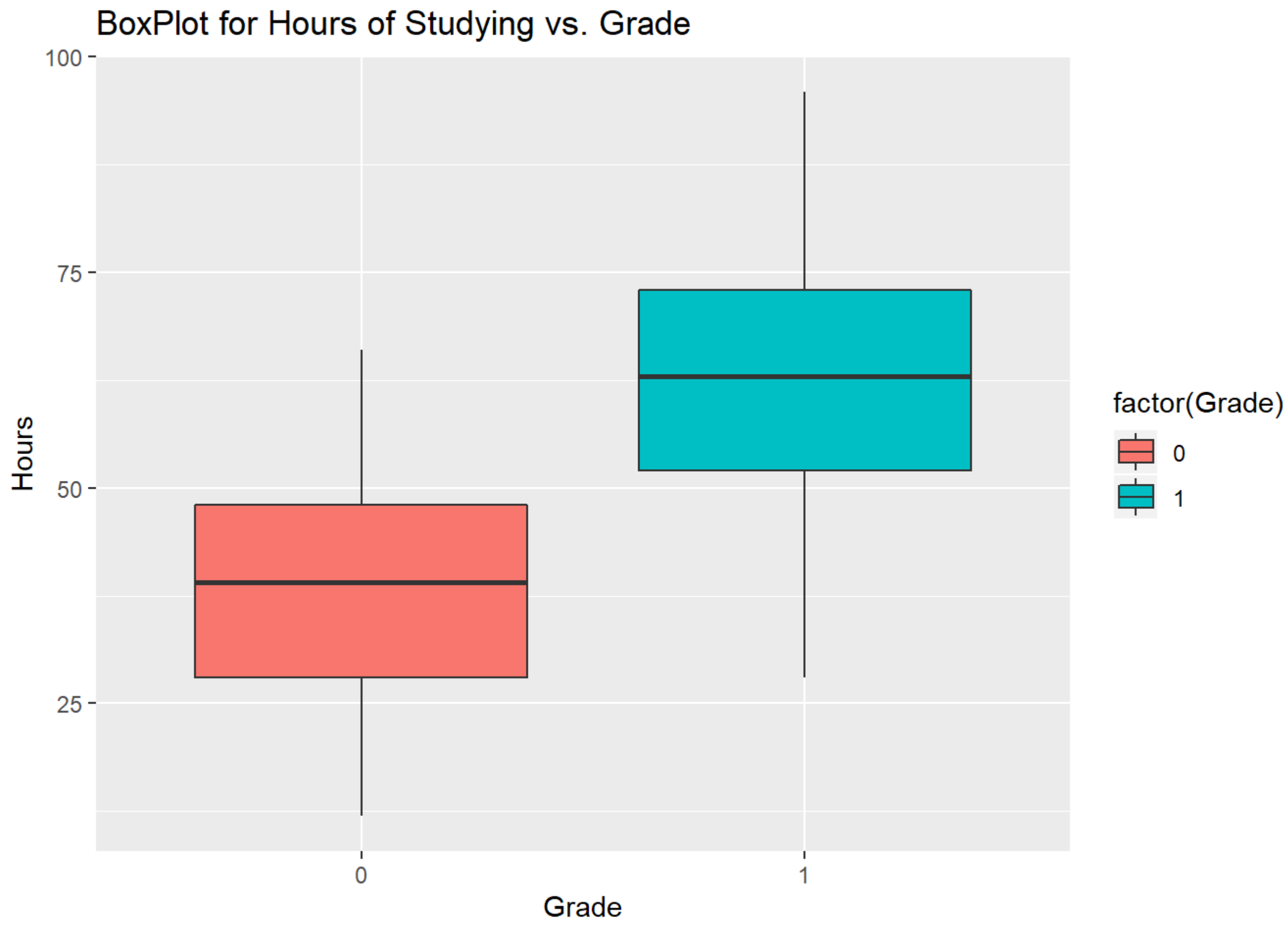
```
GradesR <- read_csv("GradesR.csv")
```

```
## Parsed with column specification:
## cols(
##   Student = col_double(),
##   Grade = col_double(),
##   Hours = col_double()
## )
```

Boxplot - Hours vs Grades

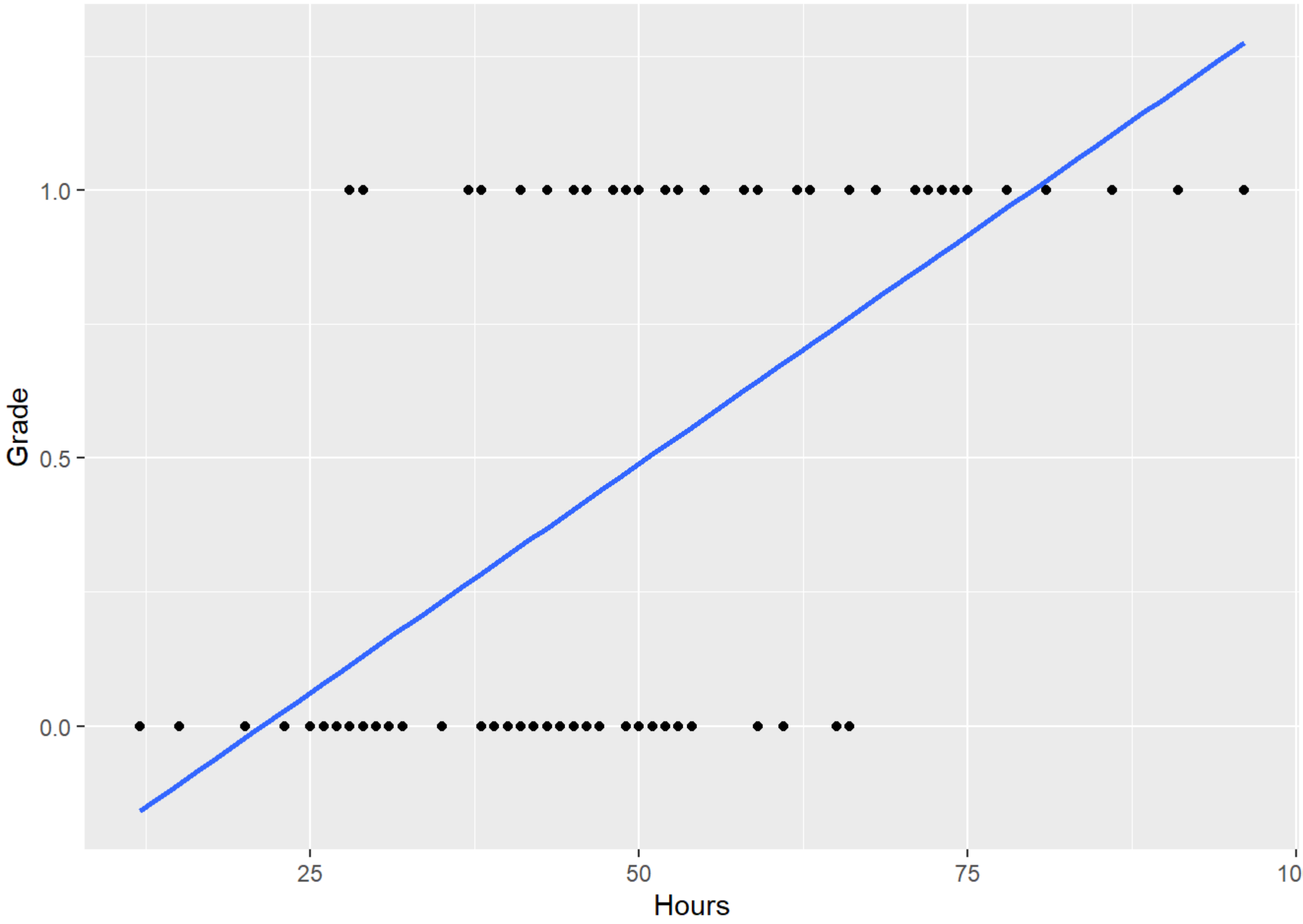
```
# Used factor function to convert Grade to a categorical variable

ggplot(data=GradesR, aes(x=factor(Grade), y = Hours, fill=factor(Grade))) + geom_boxplot() + ggtitle("BoxPlot for Hours of Studying vs. Grade") + labs(x="Grade", y="Hours")
```



Fitting linear model for Grade vs. Hours

```
ggplot(GradesR, aes(x=Hours, y=Grade)) + geom_point() +
  scale_colour_hue(l=50) +
  geom_smooth(method=lm,   # Add linear regression lines
              se=FALSE,    # Don't add shaded confidence region
              fullrange=TRUE)
```



```
a.lm <- lm(formula = Grade ~ Hours, data = GradesR)
summary(a.lm)
```

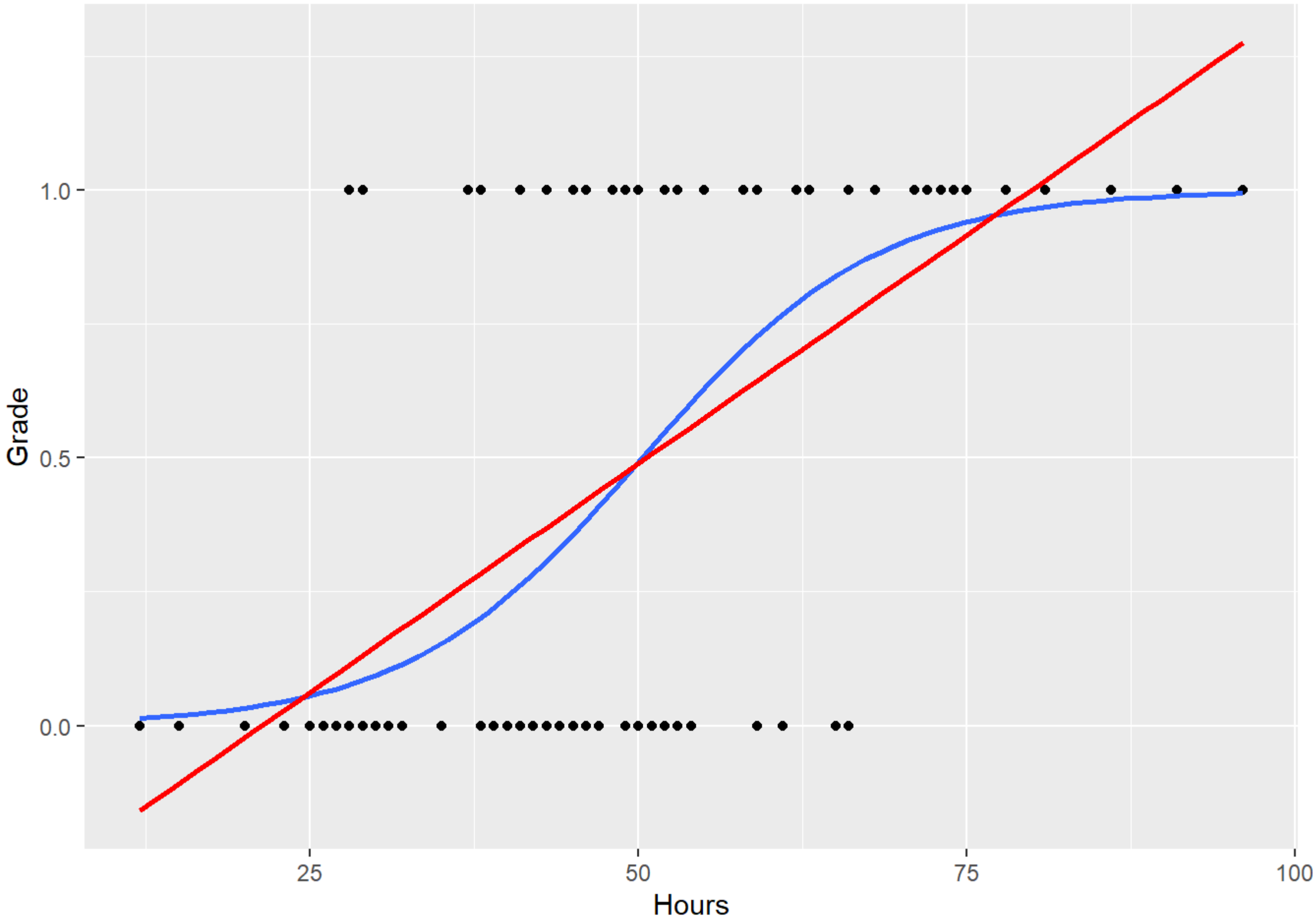
```
## 
## Call:
## lm(formula = Grade ~ Hours, data = GradesR)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.7630 -0.3060 -0.0284  0.2883  0.8862 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.364520   0.111263  -3.276  0.00146 ** 
## Hours        0.017084   0.002084   8.197 9.61e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.3889 on 98 degrees of freedom
## Multiple R-squared:  0.4068, Adjusted R-squared:  0.4007 
## F-statistic:  67.2 on 1 and 98 DF,  p-value: 9.606e-13
```

```
anova(a.lm)
```

```
## Analysis of Variance Table
## 
## Response: Grade
##           Df Sum Sq Mean Sq F value    Pr(>F)    
## Hours      1 10.165 10.1651  67.196 9.606e-13 ***
## Residuals 98 14.825  0.1513                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Comparing performance of linear model (Red) with Logistic regression model (Blue)

```
ggplot(GradesR, aes(x=Hours, y=Grade)) + geom_point() +
# add logit curve
  stat_smooth(method="glm", method.args=list(family="binomial"), se=FALSE) +
#add the regression line
    geom_smooth(method=lm,  color="red", # Add linear regression lines
              se=FALSE,    # Don't add shaded confidence region
              fullrange=TRUE)
```



Default dataset: ISLR library

```
Default_df <- Default
glimpse(Default_df)
```

```
## Observations: 10,000
## Variables: 4
## $ default <fct> No, No, No, No, No, No, No, No, No, No, No, No, No, No...
## $ student <fct> No, Yes, No, No, No, Yes, No, Yes, No, No, Yes, Yes, N...
## $ balance <dbl> 729.5265, 817.1804, 1073.5492, 529.2506, 785.6559, 919...
## $ income  <dbl> 44361.625, 12106.135, 31767.139, 35704.494, 38463.496,...
```

```
Default_df <- Default_df %>% mutate(dft_cat = ifelse(default=="Yes",1,0),
                      stdt_cat = ifelse(student=="Yes",1,0))
```

Logistic Model 1

```
Model1 <- glm(dft_cat ~ 1 , data = Default_df, family = "binomial")
summary(Model1)
```

```
##
## Call:
## glm(formula = dft_cat ~ 1, family = "binomial", data = Default_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.2603  -0.2603  -0.2603  -0.2603   2.6085
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.36833    0.05574  -60.43   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 2920.6  on 9999  degrees of freedom
## AIC: 2922.6
##
## Number of Fisher Scoring iterations: 6
```

```
Default_df %>% group_by(default) %>% summarise(number_of_datapoints=n())
```

```
## # A tibble: 2 x 2
##   default number_of_datapoints
##   <fct>                  <int>
## 1 No                      9667
## 2 Yes                      333
```

Logistic Model 2

```
Model2 <- glm(dft_cat ~ stdt_cat , data = Default_df, family = "binomial")
summary(Model2)
```

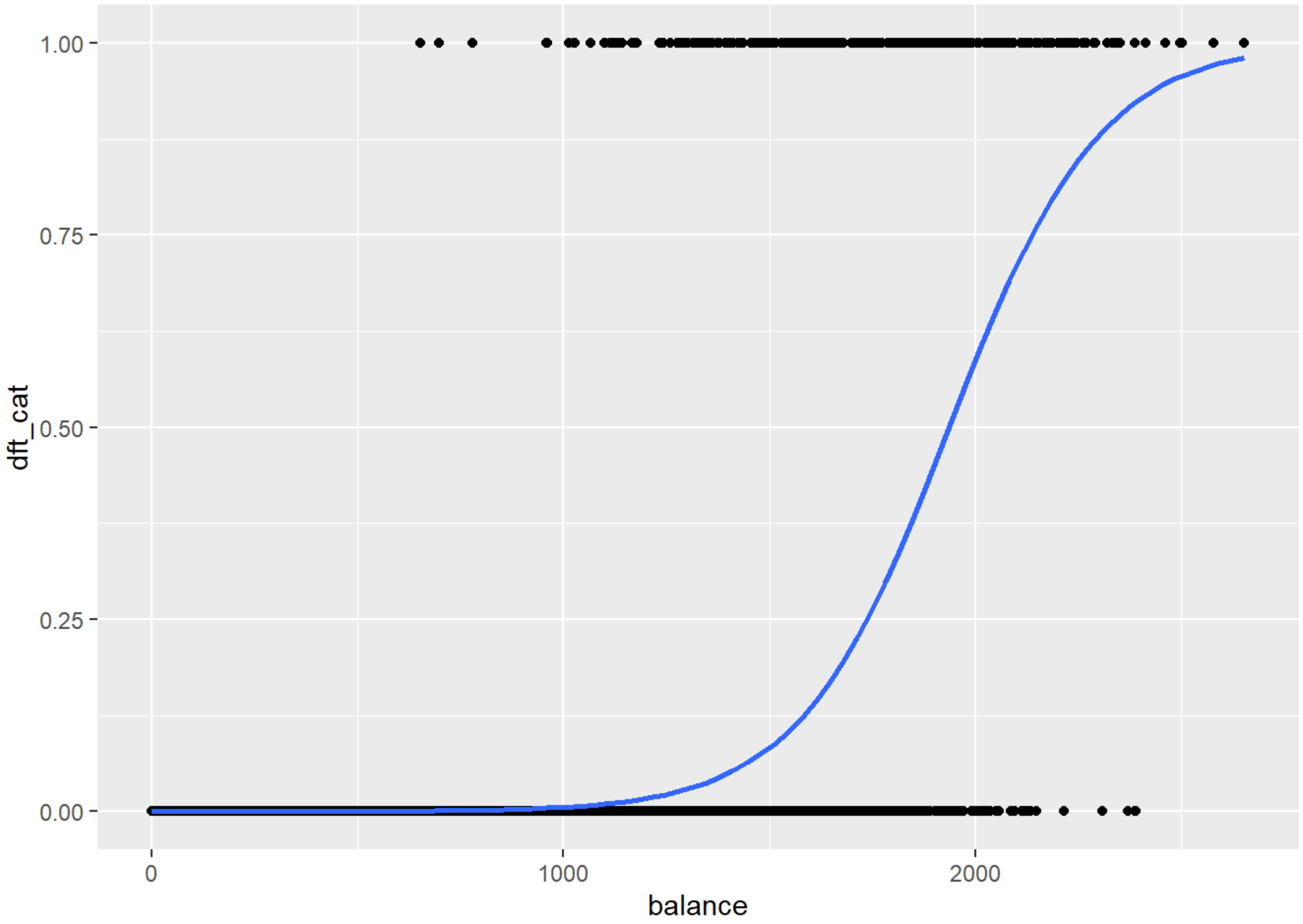
```
## 
## Call:
## glm(formula = dft_cat ~ stdt_cat, family = "binomial", data = Default_df)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -0.2970  -0.2970  -0.2434  -0.2434   2.6585  
## 
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -3.50413    0.07071  -49.55  < 2e-16 ***
## stdt_cat     0.40489    0.11502    3.52 0.000431 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 2908.7  on 9998  degrees of freedom
## AIC: 2912.7
## 
## Number of Fisher Scoring iterations: 6
```

Logistic Model 3

```
Model3 <- glm(dft_cat ~ balance , data = Default_df, family = "binomial")
summary(Model3)
```

```
## 
## Call:
## glm(formula = dft_cat ~ balance, family = "binomial", data = Default_df)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.2697  -0.1465  -0.0589  -0.0221   3.7589  
## 
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)    
## (Intercept) -1.065e+01  3.612e-01  -29.49   <2e-16 ***
## balance      5.499e-03  2.204e-04   24.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1596.5  on 9998  degrees of freedom
## AIC: 1600.5
## 
## Number of Fisher Scoring iterations: 8
```

```
ggplot(Default_df, aes(x=balance, y=dft_cat)) + geom_point() + stat_smooth(method="glm", method.
args=list(family="binomial"), se=FALSE)
```



Logistic Model 4

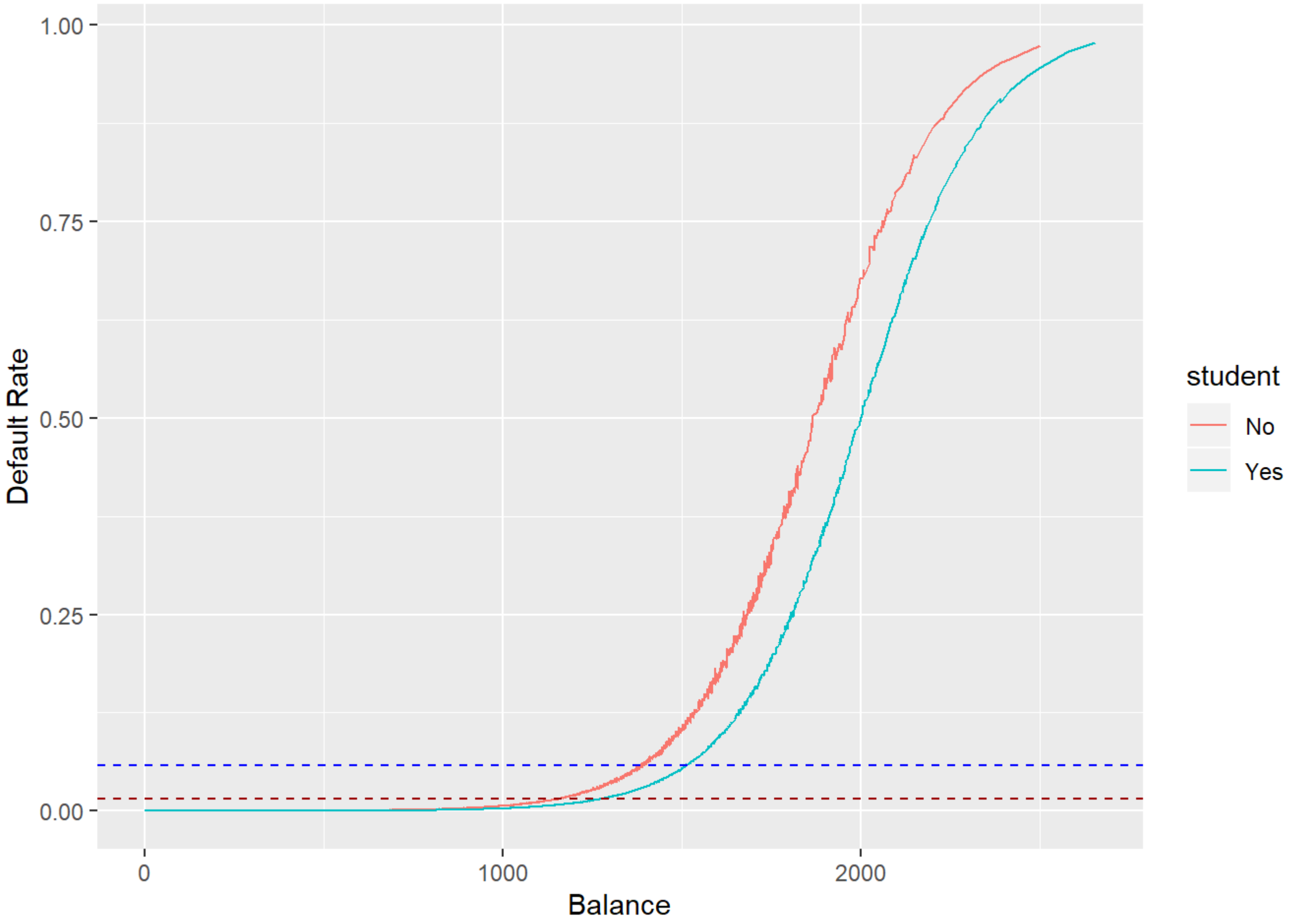
```
Model4 <- glm(dft_cat ~ balance + income + stdt_cat, data = Default_df, family = "binomial")
summary(Model4)
```

```
##
## Call:
## glm(formula = dft_cat ~ balance + income + stdt_cat, family = "binomial",
##     data = Default_df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4691  -0.1418  -0.0557  -0.0203   3.7383
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.087e+01  4.923e-01 -22.080  < 2e-16 ***
## balance      5.737e-03  2.319e-04  24.738  < 2e-16 ***
## income       3.033e-06  8.203e-06   0.370  0.71152
## stdt_cat    -6.468e-01  2.363e-01  -2.738  0.00619 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2920.6  on 9999  degrees of freedom
## Residual deviance: 1571.5  on 9996  degrees of freedom
## AIC: 1579.5
##
## Number of Fisher Scoring iterations: 8
```

Predicting default using Model 4

```
Default_df <- Default_df %>%
  mutate(pred_prob_model4 = predict(Model4, newdata = ., type = "response")) %>%
  mutate(pred_outcome_model4 = ifelse(pred_prob_model4 >= 0.5,1,0))
```

```
ggplot(data=Default_df, aes(x=balance, y=pred_prob_model4, group=student, colour=student)) +
  geom_line() +
  geom_hline(aes(yintercept=0.058), colour="blue", linetype="dashed")+
  geom_hline(aes(yintercept=0.015), colour="#990000", linetype="dashed") +
  labs(x="Balance", y="Default Rate")
```



Confusion Matrix

```
xtabs(~dft_cat + pred_outcome_model4, data = Default_df)
```

```
##        pred_outcome_model4
## dft_cat    0    1
##       0 9627   40
##       1  228  105
```

```
tally(group_by(Default_df,dft_cat,pred_outcome_model4))
```

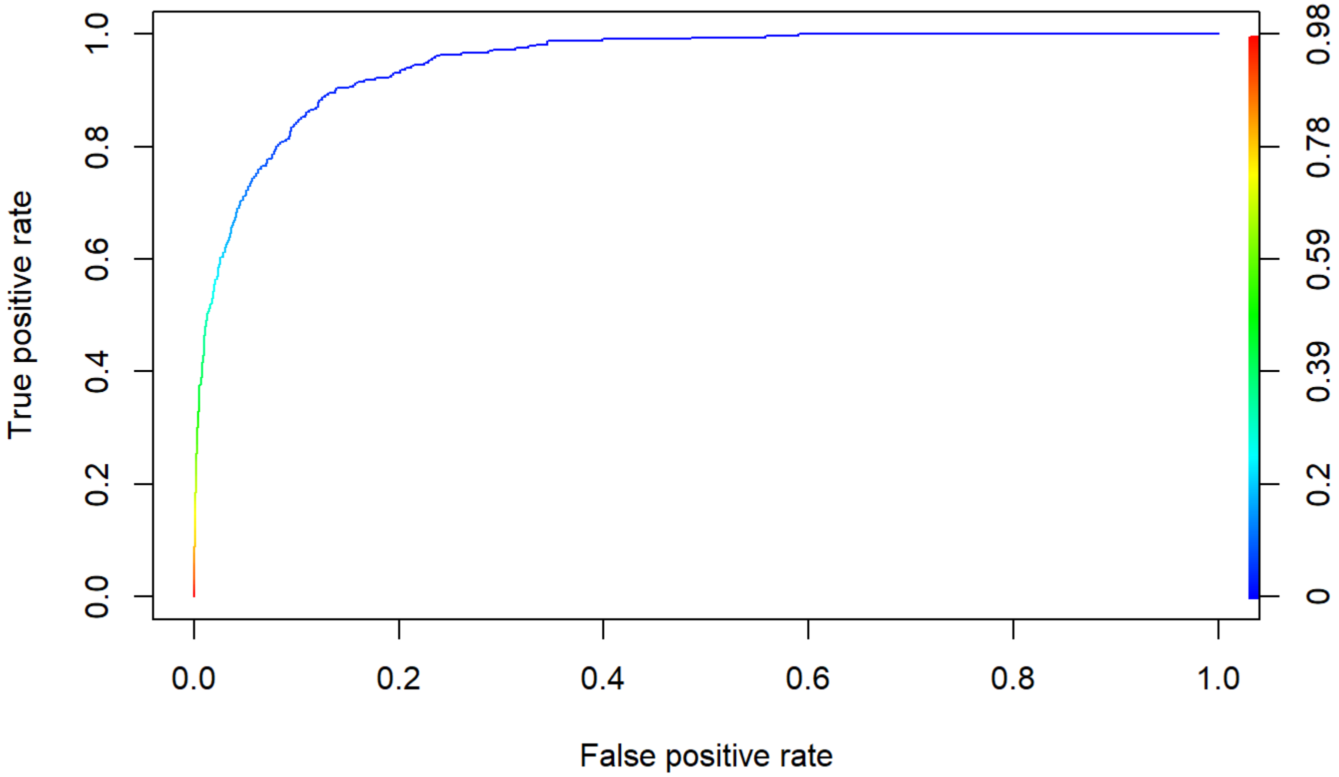
```
## # A tibble: 4 x 3
## # Groups:   dft_cat [2]
##   dft_cat pred_outcome_model4     n
##     <dbl>               <dbl> <int>
## 1       0                   0  9627
## 2       0                   1    40
## 3       1                   0   228
## 4       1                   1   105
```

ROC (Reciever Operating Characteristics) curve

```
pred <- prediction(Default_df$pred_prob_model4,Default_df$dft) # create a prediction object in R
class(pred)
```

```
## [1] "prediction"
## attr(,"package")
## [1] "ROCR"
```

```
perf <- performance(pred, "tpr", "fpr") # tpr and fpr are true and false positive rates
plot(perf, colorize=T)
```



Area under curve of ROC curve

```
auc.perf <-  performance(pred, measure = "auc")
auc.perf@y.values
```

```
## [[1]]
## [1] 0.9495581
```

Additional Reading:

You can refer this link to get more understanding about Sensitivity, Specificity and ROC curve: https://www.medcalc.org/manual/roc-curves.php (https://www.medcalc.org/manual/roc-curves.php)