Assignment 3

CMSC462

2024-10-07

```
#1
#Note: Add a categorical variable to the dataset for our multiple regression model,
#and see if it makes a statistically significant improvement to the model or not,

library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(readxl)
library(car)
```

```
## Loading required package: carData
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
Data <- read_excel("CovidMortality.xlsx")
CovidData <- data.frame(Data)

head(CovidData)
```

```
##        State Confirmed Deaths Population   Area Healthcare.Accessibility
## 1    Alabama    147153   2488    4903185  52420                      Low
## 2     Alaska      7004     45     731545 665384                      Low
## 3    Arizona    215284   5525    7278717 113990                 Moderate
## 4   Arkansas     77963   1229    3017804  53179                 Moderate
## 5 California    796436  15291   39512223 163695                 Moderate
## 6   Colorado     66649   2030    5758736 104094                     High
##   Political.Affiliation
## 1                   Red
## 2                   Red
## 3                   Red
## 4                   Red
## 5                   Red
## 6                  Blue
```

```
cat("For this model, the states will be divided between red and blue. The reason
    this is chosen is because of the varying policies and beliefs of the political
    parties during the COVID-19 pandemic, which can be easily done since there is
    already a Political.Affiliation categorical variable in the dataset.
    Differences between Red and Blue states will be represented in the model
    estimate.")
```

```
## For this model, the states will be divided between red and blue. The reason
##     this is chosen is because of the varying policies and beliefs of the political
##     parties during the COVID-19 pandemic, which can be easily done since there is
##     already a Political.Affiliation categorical variable in the dataset.
##     Differences between Red and Blue states will be represented in the model
##     estimate.
```

```
  #Split dataset into red and blue
#RedData <- CovidData[CovidData$Political.Affiliation == "Red",]
#BlueData  <- CovidData[CovidData$Political.Affiliation == "Blue",]

#RedModel <- lm(Deaths ~ Confirmed+Population+Area+Healthcare.Accessibility,data=RedData)
#summary(RedModel)

#BlueModel <- lm(Deaths ~ Confirmed+Population+Area+Healthcare.Accessibility,data=BlueData)
#summary(BlueModel)

#Create first model with all independent variables
Model <- lm(Deaths~Confirmed+Population+Area+Healthcare.Accessibility+Political.Affiliation,
            data = CovidData)
summary(Model)
```

```
##
## Call:
## lm(formula = Deaths ~ Confirmed + Population + Area + Healthcare.Accessibility +
##     Political.Affiliation, data = CovidData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6409.1 -1710.5  -321.6  1260.3 16619.7
##
## Coefficients:
##                                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)                          3.262e+03  1.474e+03   2.213   0.0344 *
## Confirmed                         3.019e-02  1.400e-02   2.156   0.0390 *
## Population                       -1.994e-05  3.276e-04  -0.061   0.9519
## Area                             -8.177e-04  6.682e-03  -0.122   0.9034
## Healthcare.AccessibilityLow      -1.575e+03  2.141e+03  -0.736   0.4674
## Healthcare.AccessibilityModerate -1.474e+03  1.744e+03  -0.845   0.4044
## Political.AffiliationRed         -3.208e+03  1.672e+03  -1.919   0.0643 .
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4124 on 31 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.6496, Adjusted R-squared:  0.5817
## F-statistic: 9.577 on 6 and 31 DF,  p-value: 5.68e-06
```

```
cat("From the summary we can see that the model is very inaccurate and not a good model
    for Deaths as the dependent variable from the given indepdendent variables. Most predictors
    fail to be statistically significant predictors for 'Deaths', only 'Confirmed'
    has a p-value less than .05 at 0.0390, and PoliticalAffiliation is very close at 0.0643.
    All other predictors have very high p-values.
    However, the F-value is very low at 5.68e-06, which does imply that the model is statistically
    significant, and at least one predictor is a good predictor for the depdendent variable,
    reinforcing 'Confirmed.' The Adjusted R-Square not good, at 0.5817 it is much less than 0.7,
    meaning that we can expect ~58.17 percent of the variance in 'Deaths' to be explained by the
    variance  of the predictors in the model, a good R-Square ranges from .7-.9")
```

```
## From the summary we can see that the model is very inaccurate and not a good model
##     for Deaths as the dependent variable from the given indepdendent variables. Most predictors
##     fail to be statistically significant predictors for ’Deaths’, only ’Confirmed’
##     has a p-value less than .05 at 0.0390, and PoliticalAffiliation is very close at 0.0643.
##     All other predictors have very high p-values.
##     However, the F-value is very low at 5.68e-06, which does imply that the model is statistically
##     significant, and at least one predictor is a good predictor for the depdendent variable,
##     reinforcing ’Confirmed.’ The Adjusted R-Square not good, at 0.5817 it is much less than 0.7,
##     meaning that we can expect ~58.17 percent of the variance in ’Deaths’ to be explained by the
##     variance  of the predictors in the model, a good R-Square ranges from .7-.9
```

```
cat("From this summary it is important to see that Population and Area are very very statistically
    insignificant with p-values of .9519 and .9034, repspectively. This strongly supports that
    neither are good predictors for the dependent variable 'Deaths.' Intuitively, this makes
    logical sense since population and area alone for states doesn't tell much about a state's
    susceptibility to a wide-spread pandemic. To adjust the model, we can add a new variable
    of population density, which is the Population / Area, then add it to the model to keep
    the Population and Area data relevant, but in a different form.")
```

```
## From this summary it is important to see that Population and Area are very very statistically
##     insignificant with p-values of .9519 and .9034, repspectively. This strongly supports that
##     neither are good predictors for the dependent variable ’Deaths.’ Intuitively, this makes
##     logical sense since population and area alone for states doesn’t tell much about a state’s
##     susceptibility to a wide-spread pandemic. To adjust the model, we can add a new variable
##     of population density, which is the Population / Area, then add it to the model to keep
##     the Population and Area data relevant, but in a different form.
```

```
#Density = Population/Area
CovidData$PopDensity <- CovidData$Population/CovidData$Area
#Rerun model with Population Density
Model <- lm(Deaths~Confirmed+PopDensity+Healthcare.Accessibility+Political.Affiliation,
            data = CovidData)
summary(Model)
```

```
##
## Call:
## lm(formula = Deaths ~ Confirmed + PopDensity + Healthcare.Accessibility +
##     Political.Affiliation, data = CovidData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6586.8 -1744.3  -259.9  1228.3 16606.8
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       3.230e+03  1.393e+03   2.319   0.0269 *
## Confirmed                         2.938e-02  3.918e-03   7.499 1.55e-08 ***
## PopDensity                       -6.204e-02  4.202e-01  -0.148   0.8835
## Healthcare.AccessibilityLow      -1.536e+03  2.093e+03  -0.734   0.4682
## Healthcare.AccessibilityModerate -1.441e+03  1.741e+03  -0.828   0.4140
## Political.AffiliationRed         -3.293e+03  1.653e+03  -1.993   0.0549 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4059 on 32 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.6496, Adjusted R-squared:  0.5948
## F-statistic: 11.86 on 5 and 32 DF,  p-value: 1.529e-06
```

```
cat("From this summary we can see that the Adjusted R-Square did improve marginally,
    up to 0.5948 from 0.5817, and the p-value of Population and Area, combined, improved,
    but PopDensity is still widely statistically insignificant at a p-value of 0.8835")
```

```
## From this summary we can see that the Adjusted R-Square did improve marginally,
##     up to 0.5948 from 0.5817, and the p-value of Population and Area, combined, improved,
##     but PopDensity is still widely statistically insignificant at a p-value of 0.8835
```

```
cat("For my own categorical variable, I will import a dataset of vaccinations rates per state
    for COVID, and define my own thresholds of vaccination rates to determine if a state has
    Low/Medium/High vaccination rates, then add to the regression model to rerun.")
```

```
## For my own categorical variable, I will import a dataset of vaccinations rates per state
##     for COVID, and define my own thresholds of vaccination rates to determine if a state has
##     Low/Medium/High vaccination rates, then add to the regression model to rerun.
```

```
#Import dataset from https://data.cms.gov/provider-data/dataset/avax-cv19
#For vaccination percentages for states in the US
vaccineData <- read_csv(file =
  "C:/Users/criss/Desktop/CMSC462/Assignments/[10-20]Assigment 3/NH_CovidVaxAverages.csv")
```

```
## Rows: 54 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): State, Date vaccination data last updated
## dbl (2): Percent of residents who are up-to-date on their vaccines, Percent ...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
head(vaccineData)
```

```
## # A tibble: 6 x 4
##   State Percent of residents who~1 Percent of staff who~2 Date vaccination dat~3
##   <chr>                      <dbl>                  <dbl> <chr>
## 1 US                          28.2                    9.3 09/29/2024
## 2 AK                          42.4                   13.5 09/29/2024
## 3 AL                          25.2                    5.1 09/29/2024
## 4 AR                          19.3                    4.2 09/29/2024
## 5 AZ                          18.4                    6   09/29/2024
## 6 CA                          37.4                   15.9 09/29/2024
## # i abbreviated names:
## #   1: ‘Percent of residents who are up-to-date on their vaccines‘,
## #   2: ‘Percent of staff who are up-to-date on their vaccines‘,
## #   3: ‘Date vaccination data last updated‘
```

```
#Categorize rates of vaccinations, split by 20 and 50 percentiles to low/medium/high
#Mutate dataset to add new attribute
vaccineData <- vaccineData %>% mutate(vaccineRate =
  cut(`Percent of residents who are up-to-date on their vaccines`,
  breaks = quantile(`Percent of residents who are up-to-date on their vaccines`,
  probs = c(0, .20, 0.50, 1)),
  labels = c("Low", "Medium", "High"),include.lowest = TRUE))

head(vaccineData)
```

```
## # A tibble: 6 x 5
##   State Percent of residents who~1 Percent of staff who~2 Date vaccination dat~3
##   <chr>                      <dbl>                  <dbl> <chr>
## 1 US                          28.2                    9.3 09/29/2024
## 2 AK                          42.4                   13.5 09/29/2024
## 3 AL                          25.2                    5.1 09/29/2024
## 4 AR                          19.3                    4.2 09/29/2024
## 5 AZ                          18.4                    6   09/29/2024
## 6 CA                          37.4                   15.9 09/29/2024
## # i abbreviated names:
## #   1: ‘Percent of residents who are up-to-date on their vaccines‘,
## #   2: ‘Percent of staff who are up-to-date on their vaccines‘,
## #   3: ‘Date vaccination data last updated‘
## # i 1 more variable: vaccineRate <fct>
```

```
cat("Since the new dataset has state abbreviations instead of full names, I will
    need to create a mapping between the 2 datasets in order to merge the new
    categorical variable into the model.")
```

```
## Since the new dataset has state abbreviations instead of full names, I will
##     need to create a mapping between the 2 datasets in order to merge the new
##     categorical variable into the model.
```

```r
mapping <- data.frame(
  Abbreviation = c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "DC", "FL", "GA",
                   "HI", "ID", "IL", "IN", "IA",
                   "KS", "KY", "LA", "ME", "MD", "MA", "MI", "MN", "MS", "MO", "MT",
                   "NE", "NV", "NH", "NJ", "NM",
                   "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC", "SD", "TN",
                   "TX", "UT", "VT", "VA", "WA",
                   "WV", "WI", "WY"),
  State = c("Alabama", "Alaska", "Arizona", "Arkansas", "California", "Colorado",
            "Connecticut", "Delaware", "District of Columbia", "Florida", "Georgia",
            "Hawaii", "Idaho", "Illinois", "Indiana", "Iowa", "Kansas", "Kentucky",
            "Louisiana", "Maine", "Maryland", "Massachusetts", "Michigan", "Minnesota",
            "Mississippi", "Missouri", "Montana", "Nebraska", "Nevada", "New Hampshire",
            "New Jersey", "New Mexico", "New York", "North Carolina", "North Dakota",
            "Ohio", "Oklahoma","Oregon", "Pennsylvania", "Rhode Island", "South Carolina",
            "South Dakota", "Tennessee", "Texas", "Utah", "Vermont", "Virginia",
            "Washington", "West Virginia", "Wisconsin","Wyoming")
)

#Merge mapping and full names to the vaccine dataset for each entry
vaccineData <- merge(vaccineData, mapping, by.x="State", by.y="Abbreviation",all.x=TRUE)
#Replace State column with the full names of states from State.y
vaccineData <- mutate(vaccineData, State = State.y)
temp <- vaccineData[,c("State", "vaccineRate")]
head(vaccineData)
```

```
##        State Percent of residents who are up-to-date on their vaccines
## 1     Alaska                                                      42.4
## 2    Alabama                                                      25.2
## 3   Arkansas                                                      19.3
## 4    Arizona                                                      18.4
## 5 California                                                      37.4
## 6   Colorado                                                      30.7
##   Percent of staff who are up-to-date on their vaccines
## 1                                                  13.5
## 2                                                   5.1
## 3                                                   4.2
## 4                                                   6.0
## 5                                                  15.9
## 6                                                  13.0
##   Date vaccination data last updated vaccineRate    State.y
## 1                         09/29/2024        High     Alaska
## 2                         09/29/2024      Medium    Alabama
## 3                         09/29/2024         Low   Arkansas
## 4                         09/29/2024         Low    Arizona
## 5                         09/29/2024        High California
## 6                         09/29/2024        High   Colorado
```

```r
#Merge vaccineRate to CovidData dataset as a new column
CovidData <- merge(CovidData, temp, by="State", all.x=TRUE)
CovidData$vaccineRate <- as.factor(CovidData$vaccineRate)
head(CovidData)
```

```
##        State Confirmed Deaths Population   Area Healthcare.Accessibility
## 1    Alabama    147153   2488    4903185  52420                      Low
## 2     Alaska      7004     45     731545 665384                      Low
## 3    Arizona    215284   5525    7278717 113990                 Moderate
## 4   Arkansas     77963   1229    3017804  53179                 Moderate
## 5 California    796436  15291   39512223 163695                 Moderate
## 6   Colorado     66649   2030    5758736 104094                     High
##   Political.Affiliation PopDensity vaccineRate
## 1                   Red  93.536532      Medium
## 2                   Red   1.099433        High
## 3                   Red  63.853996         Low
## 4                   Red  56.748040         Low
## 5                   Red 241.377092        High
## 6                  Blue  55.322459        High
```

```r
#Rerun model with vaccineRate categorical variable
Model <- lm(Deaths~Confirmed+PopDensity+Healthcare.Accessibility+Political.Affiliation+
              vaccineRate, data = CovidData)
summary(Model)
```

```
##
## Call:
## lm(formula = Deaths ~ Confirmed + PopDensity + Healthcare.Accessibility +
##     Political.Affiliation + vaccineRate, data = CovidData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6495.9 -1756.6    53.6  1269.2 16537.5
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       3.536e+03  2.663e+03   1.328   0.1943
## Confirmed                         2.902e-02  4.116e-03   7.051 7.74e-08 ***
## PopDensity                       -1.059e-01  4.422e-01  -0.239   0.8124
## Healthcare.AccessibilityLow      -1.217e+03  2.273e+03  -0.535   0.5963
## Healthcare.AccessibilityModerate -1.270e+03  1.859e+03  -0.683   0.4998
## Political.AffiliationRed         -3.506e+03  1.794e+03  -1.954   0.0601 .
## vaccineRateMedium                -8.306e+02  2.283e+03  -0.364   0.7185
## vaccineRateHigh                  -5.888e+01  2.163e+03  -0.027   0.9785
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4175 on 30 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.6525, Adjusted R-squared:  0.5714
## F-statistic: 8.047 on 7 and 30 DF,  p-value: 1.677e-05
```

```r
cat("Alas, the model does not improve with this new categorical variable of vaccine
    rates from the percentages of residents who received vaccines for COVID-19 up to
    2024. The R-Squared value drops to 0.5714 from the previously calculated 0.5948,
    and the p-value for the vaccineRate predictor is very high, at 0.7185 for a
    'medium' vaccine rate, and 0.9785 for a 'high' vaccineRate.")
```

```
## Alas, the model does not improve with this new categorical variable of vaccine
##     rates from the percentages of residents who received vaccines for COVID-19 up to
##     2024. The R-Squared value drops to 0.5714 from the previously calculated 0.5948,
##     and the p-value for the vaccineRate predictor is very high, at 0.7185 for a
##     ’medium’ vaccine rate, and 0.9785 for a ’high’ vaccineRate.
```

```r
#Calculate correlation between predictors to remove from the model
correlation_matrix <- cor(CovidData[, sapply(CovidData, is.numeric)])
correlation_matrix
```

```
##              Confirmed      Deaths Population        Area  PopDensity
## Confirmed   1.00000000  0.76455249  0.9550060  0.15412370 -0.06833992
## Deaths      0.76455249  1.00000000  0.7566326  0.01692798 -0.01489674
## Population  0.95500596  0.75663259  1.0000000  0.14795734 -0.08078940
## Area        0.15412370  0.01692798  0.1479573  1.00000000 -0.15573386
## PopDensity -0.06833992 -0.01489674 -0.0807894 -0.15573386  1.00000000
```

```r
cat("If we look at the correlation matrix, we see that from the quantitative predictors,
    Population is highly correlated with other predictors, namely 'Confirmed' at a correlation
    of .955. We have already removed Population from the model and replaced it with
    PopDensity, therefore no changes needed at this time in regards to correlated predictors.
    Confirmed is highly correlated with deaths, but it is a very significant predictor
    for the model, therefore it is kept as the exception predictor.")
```

```
## If we look at the correlation matrix, we see that from the quantitative predictors,
##     Population is highly correlated with other predictors, namely ’Confirmed’ at a correlation
##     of .955. We have already removed Population from the model and replaced it with
##     PopDensity, therefore no changes needed at this time in regards to correlated predictors.
##     Confirmed is highly correlated with deaths, but it is a very significant predictor
##     for the model, therefore it is kept as the exception predictor.
```

```r
Model <- lm(Deaths~Confirmed+Political.Affiliation+vaccineRate, data = CovidData)
summary(Model)
```

```
##
## Call:
## lm(formula = Deaths ~ Confirmed + Political.Affiliation + vaccineRate,
##     data = CovidData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6794.7 -2357.0   126.5  1226.8 17187.2
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)               2.586e+03  2.290e+03   1.129   0.2671
## Confirmed                 2.902e-02  3.952e-03   7.344 1.97e-08 ***
## Political.AffiliationRed -3.816e+03  1.437e+03  -2.656   0.0121 *
## vaccineRateMedium        -7.741e+02  2.174e+03  -0.356   0.7241
## vaccineRateHigh           2.028e+02  2.047e+03   0.099   0.9217
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4024 on 33 degrees of freedom
##   (14 observations deleted due to missingness)
## Multiple R-squared:  0.6449, Adjusted R-squared:  0.6018
## F-statistic: 14.98 on 4 and 33 DF,  p-value: 4.436e-07
```

```
cat("If we remove the statistically insignificant predictors except for Confirmed,
    Political Affiliation, and the new vaccineRate, the model is in its best shape
    with an adjusted R-Square of 0.6018 and statistically significant predictors
    'Confirmed' and 'Political.Affiliation' both with p-values less than .05. If we choose
    this to be the final model, then we can describe the model as such:
    For every state there were a total of 2568 base deaths, for every confirmed
    case of COVID-19 0.02902 deaths occur, Red states expect 3816 less deaths than
    blue states, states with Medium vaccination rates expect 774.1 less deaths
    than states with Low vaccination rates, and states with High vaccination rates
    expect 202.8 more deaths than states with Medium vaccination rates. Note that
    the intercept and vaccineRates are not good predictors of Deaths because of
    their p-values.

    Equation: 2568 + .02902x1 - 3816x2 - 774.1x3 + 202.8x4
    x1 = Confirmed cases
    x2 = Red or Blue state(1 or 0, 1=Red)
    x3 = Medium Vaccine rate(1 or 0, 1=yes)
    x4 = High Vaccine rate(1 or 0, 1=yes)

    Adjusted R-Square: 0.6018
    Implies ~60.18 percent of the variance in deaths is explained by the variance
    in the selected predictors; the model is statistically significant with
    an F-value < .05, but the fit of the model is not strong.")
```

```
## If we remove the statistically insignificant predictors except for Confirmed,
##     Political Affiliation, and the new vaccineRate, the model is in its best shape
##     with an adjusted R-Square of 0.6018 and statistically significant predictors
##     ’Confirmed’ and ’Political.Affiliation’ both with p-values less than .05. If we choose
##     this to be the final model, then we can describe the model as such:
##     For every state there were a total of 2568 base deaths, for every confirmed
##     case of COVID-19 0.02902 deaths occur, Red states expect 3816 less deaths than
##     blue states, states with Medium vaccination rates expect 774.1 less deaths
##     than states with Low vaccination rates, and states with High vaccination rates
##     expect 202.8 more deaths than states with Medium vaccination rates. Note that
##     the intercept and vaccineRates are not good predictors of Deaths because of
##     their p-values.
##
##     Equation: 2568 + .02902x1 - 3816x2 - 774.1x3 + 202.8x4
##     x1 = Confirmed cases
##     x2 = Red or Blue state(1 or 0, 1=Red)
##     x3 = Medium Vaccine rate(1 or 0, 1=yes)
##     x4 = High Vaccine rate(1 or 0, 1=yes)
##
##     Adjusted R-Square: 0.6018
##     Implies ~60.18 percent of the variance in deaths is explained by the variance
##     in the selected predictors; the model is statistically significant with
##     an F-value < .05, but the fit of the model is not strong.
```

```
#2
#Draw the ROC, which is install.packages("pROC"), and then generates a graph of true positives
#divided by true positives+false negatives, and false positives divided by false positives and true neg
#install.packages("ROCR")
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(ROCR)
library(e1071)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
#Import data
Data <- read_csv(file =
  "C:/Users/criss/Desktop/CMSC462/Assignments/[10-20]Assigment 3/Lending.csv")
```

```
## Rows: 88451 Columns: 12
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): residence_property
## dbl (11): loan_default, loan_amnt, adjusted_annual_inc, pct_loan_income, dti...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

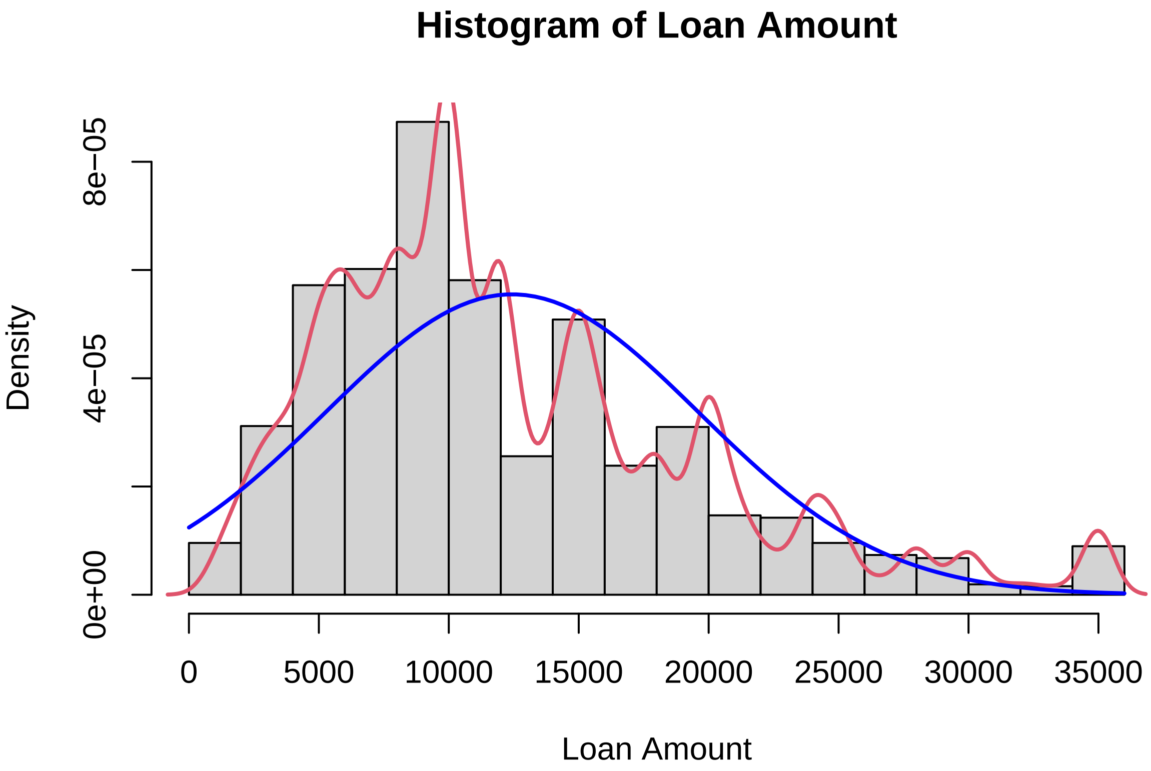
```r
head(Data)
```

```
## # A tibble: 6 x 12
##   loan_default loan_amnt adjusted_annual_inc pct_loan_income   dti
##          <dbl>     <dbl>               <dbl>           <dbl> <dbl>
## 1            1     15000               41640           0.278 23.4 
## 2            0      8000               64640           0.104  2.76
## 3            0     14000               29132           0.28  17.8 
## 4            0      4000               25280           0.08  29.0 
## 5            0     18825               28344           0.448 15.7 
## 6            0     10500               21048           0.318 13.2 
## # i 7 more variables: residence_property <chr>,
## #   months_since_first_credit <dbl>, inq_last_6mths <dbl>, open_acc <dbl>,
## #   bc_util <dbl>, num_accts_ever_120_pd <dbl>, pub_rec_bankruptcies <dbl>
```

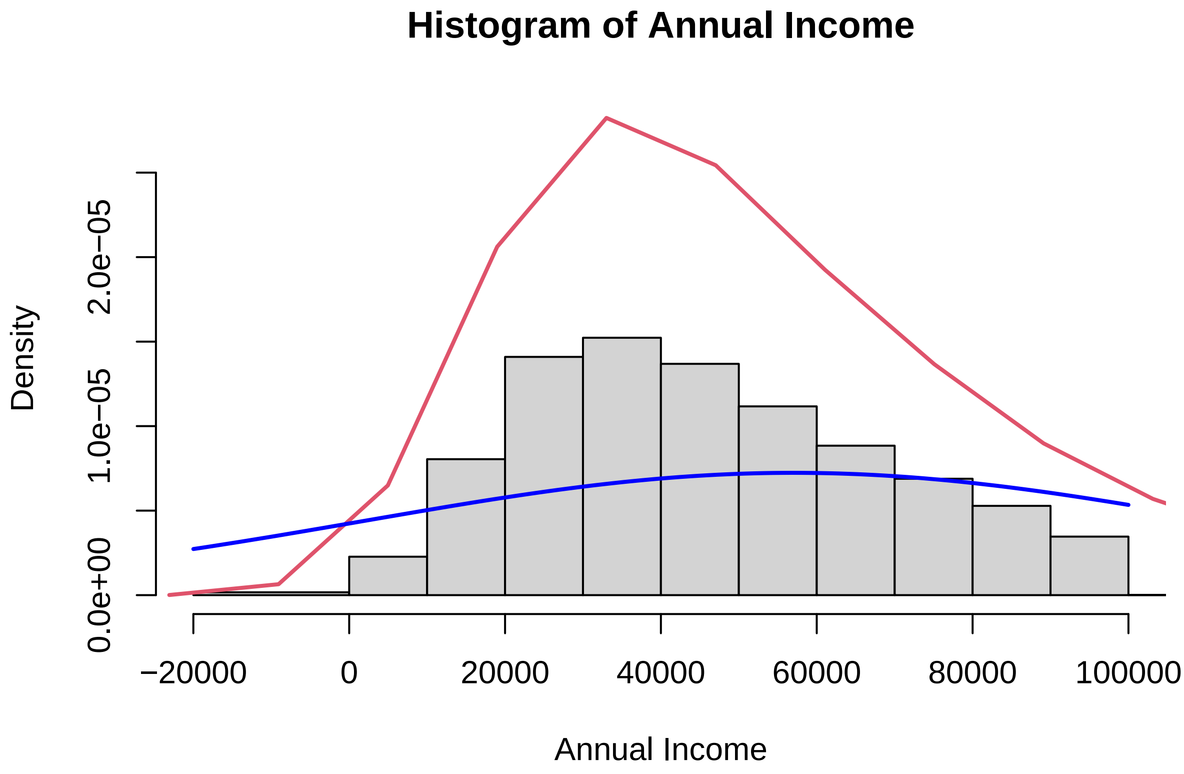
```r
#(1) - Descriptive Statistics
summary(Data)
```

```
##   loan_default      loan_amnt     adjusted_annual_inc pct_loan_income   
##  Min.   :0.0000   Min.   : 1000   Min.   : -14540     Min.   :0.002076  
##  1st Qu.:0.0000   1st Qu.: 7200   1st Qu.:  30176     1st Qu.:0.123087  
##  Median :0.0000   Median :10500   Median :  47028     Median :0.188889  
##  Mean   :0.1253   Mean   :12435   Mean   :  57014     Mean   :0.201073  
##  3rd Qu.:0.0000   3rd Qu.:16000   3rd Qu.:  71574     3rd Qu.:0.269231  
##  Max.   :1.0000   Max.   :35000   Max.   :7135346     Max.   :0.450000  
##       dti        residence_property months_since_first_credit inq_last_6mths  
##  Min.   : 0.00   Length:88451       Min.   : 36.0             Min.   :0.0000  
##  1st Qu.:11.04   Class  :character   1st Qu.:126.0             1st Qu.:0.0000  
##  Median :16.49   Mode  :character   Median :166.0             Median :0.0000  
##  Mean   :16.90                      Mean   :183.3             Mean   :0.7827  
##  3rd Qu.:22.52                      3rd Qu.:225.0             3rd Qu.:1.0000  
##  Max.   :34.99                      Max.   :750.0             Max.   :7.0000  
##     open_acc        bc_util       num_accts_ever_120_pd pub_rec_bankruptcies
##  Min.   : 1.00   Min.   :  0.00   Min.   : 0.0000       Min.   :0.00000     
##  1st Qu.: 8.00   1st Qu.: 49.30   1st Qu.: 0.0000       1st Qu.:0.00000     
##  Median :10.00   Median : 72.10   Median : 0.0000       Median :0.00000     
##  Mean   :10.87   Mean   : 66.71   Mean   : 0.3225       Mean   :0.09236     
##  3rd Qu.:13.00   3rd Qu.: 89.00   3rd Qu.: 0.0000       3rd Qu.:0.00000     
##  Max.   :62.00   Max.   :173.20   Max.   :29.0000       Max.   :7.00000     
```

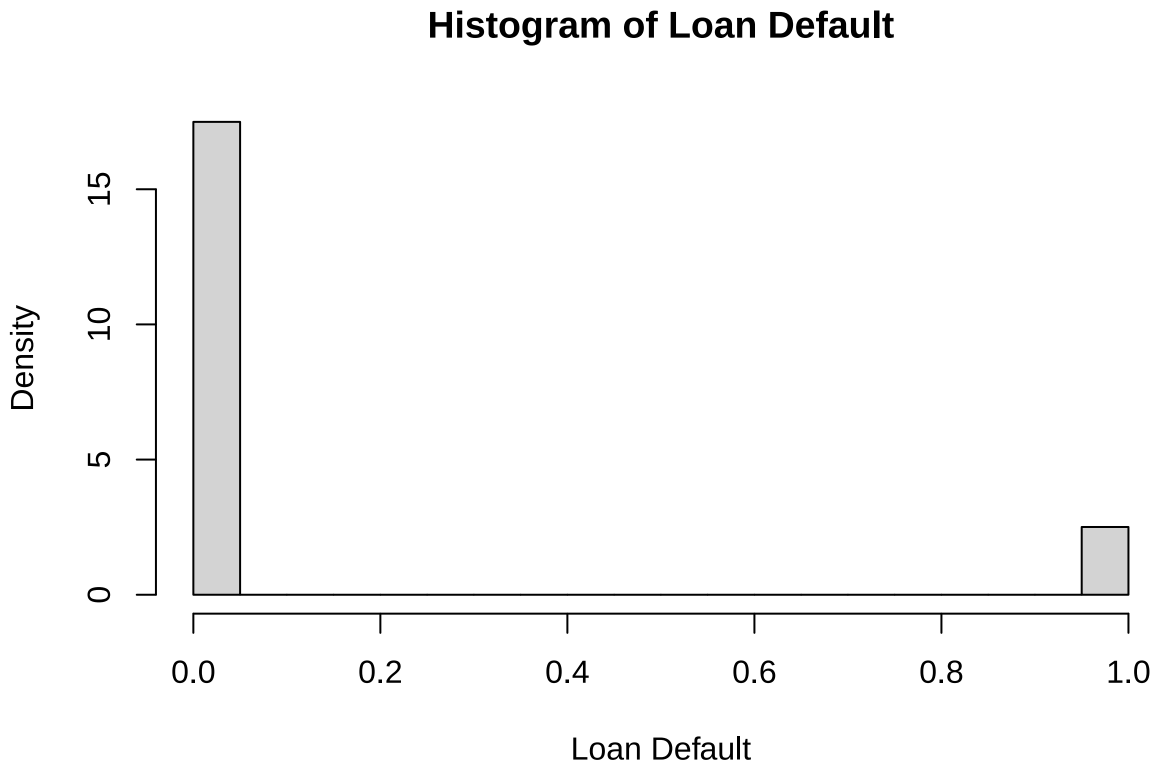
```r
#Histogram + curve of loan amount, superimposed normal curve
hist(Data$loan_amnt, probability=TRUE, main = "Histogram of Loan Amount", xlab="Loan Amount")
lines(density(Data$loan_amnt), col = 2, lwd = 2)
curve(dnorm(x, mean=mean(Data$loan_amnt), sd=sd(Data$loan_amnt)),
      lwd=2, col="blue", add=TRUE)
```



```
#Histogram + curve of adjusted annual income, superimposed normal curve
hist(Data$adjusted_annual_inc, probability=TRUE, main = "Histogram of Annual Income",
     xlab="Annual Income", xlim=c(-20000,100000),ylim=c(0,0.000028),
     breaks=c(-20000,0,10000,20000,30000,
              40000,50000,60000,70000,
              80000,90000,100000,10000000))
lines(density(Data$adjusted_annual_inc), col = 2, lwd = 2)
curve(dnorm(x, mean=mean(Data$adjusted_annual_inc), sd=sd(Data$adjusted_annual_inc)),
      lwd=2, col="blue", add=TRUE)
```



```
#Histogram of loan default
hist(Data$loan_default, probability=TRUE, main="Histogram of Loan Default",
     xlab="Loan Default")
```



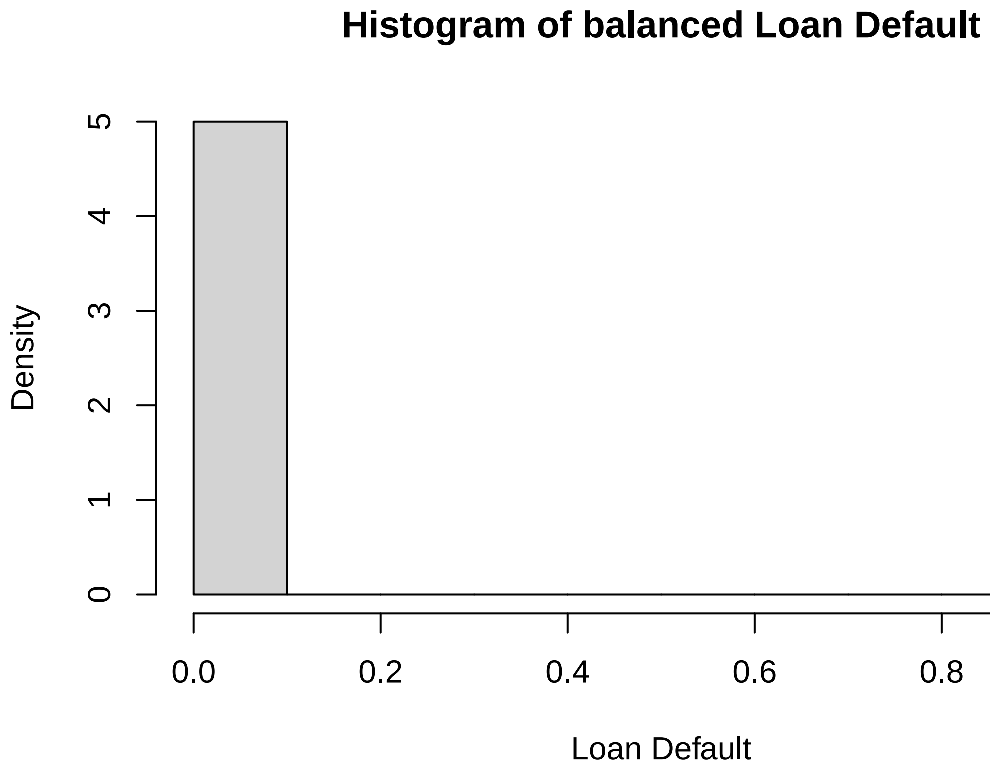
```r
cat("We can see that the data is pretty skewed from the summary statistics and the
    histograms with superimposed normal curves of the loan amounts and adjusted annual
    incomes. We can also see that the entries in the data are extremely unbalanced
    when viewing the loan_default histogram, with loan_default=0 being much more
    common that loan_default=1 for most entries. To resolve this we resample the
    data inorder to balance the entries with default=0 and default=1")
```

```
## We can see that the data is pretty skewed from the summary statistics and the
##     histograms with superimposed normal curves of the loan amounts and adjusted annual
##     incomes. We can also see that the entries in the data are extremely unbalanced
##     when viewing the loan_default histogram, with loan_default=0 being much more
##     common that loan_default=1 for most entries. To resolve this we resample the
##     data inorder to balance the entries with default=0 and default=1
```

```r
#From the dataset, take 2000 random samples where there is a default=1 and default=0
#Create a new dataset from the samples to balance the data, and use that dataset for the model
#The balanced data set will see a great fall in accuracy
set.seed(1)
Lending = rbind(sample_n(filter(Data, loan_default==1), 2000),
                sample_n(filter(Data, loan_default==0), 2000))

#Convert categorical data to factor
Lending$residence_property <- as.factor(Lending$residence_property)

hist(Lending$loan_default, probability=TRUE, main="Histogram of balanced Loan Default",
     xlab="Loan Default")
```



```
#(2) Naive Bayes Model
# Sample of 75% of the data used to train
train_ind<- sample(1:nrow(Lending), size = (0.75 * nrow(Lending)))

#Separate data into training and testing, 75% training set is excluded from test set
train <- Lending[train_ind,]
test <- Lending[-train_ind,]

#Logistic regression model
NB <- naiveBayes(loan_default ~ ., data=train)
NB
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     0     1
## 0.498 0.502
##
## Conditional probabilities:
##    loan_amnt
## Y       [,1]     [,2]
##   0 12734.61 7290.103
##   1 12163.61 7199.355
##
##    adjusted_annual_inc
## Y       [,1]     [,2]
##   0 56684.45 39357.55
##   1 48495.98 40147.52
##
##    pct_loan_income
## Y        [,1]      [,2]
##   0 0.2033369 0.1007958
##   1 0.2201587 0.1042627
##
##    dti
## Y       [,1]     [,2]
##   0 16.67671 7.590641
##   1 17.97578 7.546656
##
##    residence_property
## Y         Own      Rent
##   0 0.5917001 0.4082999
##   1 0.4840637 0.5159363
##
##    months_since_first_credit
## Y       [,1]     [,2]
##   0 184.1124 87.98647
##   1 174.5558 83.38455
##
##    inq_last_6mths
## Y        [,1]      [,2]
##   0 0.7409639 0.9644564
##   1 0.9555113 1.0709139
##
##    open_acc
## Y       [,1]     [,2]
##   0 10.91232 4.508793
##   1 10.99070 4.622340
##
##    bc_util
## Y       [,1]     [,2]
##   0 66.74752 26.60818
##   1 70.75100 24.66796
##
##    num_accts_ever_120_pd
## Y        [,1]      [,2]
##   0 0.3085676 0.8555208
##   1 0.3014608 0.9219136
##
##    pub_rec_bankruptcies
## Y         [,1]      [,2]
##   0 0.09036145 0.2914283
##   1 0.09628154 0.3104377
```

```r
#Predictions from Naive Bayes model using test dataset
predictions <- predict(NB, newdata = test)

#Confusion matrix of results
confusion_matrix <- table(predictions, test$loan_default)
confusion_matrix
```

```
##
## predictions   0   1
##           0 298 197
##           1 208 297
```

```r
confusionMatrix(confusion_matrix)
```

```
## Confusion Matrix and Statistics
##
##
## predictions   0   1
##           0 298 197
##           1 208 297
##
##                Accuracy : 0.595
##                  95% CI : (0.5638, 0.6256)
##     No Information Rate : 0.506
##     P-Value [Acc > NIR] : 9.78e-09
##
##                   Kappa : 0.1901
##
##  Mcnemar's Test P-Value : 0.6193
##
##             Sensitivity : 0.5889
##             Specificity : 0.6012
##          Pos Pred Value : 0.6020
##          Neg Pred Value : 0.5881
##              Prevalence : 0.5060
##          Detection Rate : 0.2980
##    Detection Prevalence : 0.4950
##       Balanced Accuracy : 0.5951
##
##        'Positive' Class : 0
##
```

```r
cat("297 True positives
    298 True negatives
    208 False positives
    197 False negatives")
```

```
## 297 True positives
##     298 True negatives
##     208 False positives
##     197 False negatives
```

```
    #predictionsRaw <- predict(NB, newdata = test, type = "raw")
    #predictionsRaw

#Calculate accuracy from CM
NB_accuracy <- mean(predictions == test$loan_default)
NB_accuracy
```

```
## [1] 0.595
```

```
cat("This Naive Bayes model classifies whether or not a person obtains a loan in
    the categorical variable loan_default, where 1=loan received and 0=no loan given.
    From the various attrbutes of each person, the model predicts if a person receives
    a loan, classifying them. The accuracy of the model is very low at 0.595, due to
    our balancing of the original dataset.")
```

```
## This Naive Bayes model classifies whether or not a person obtains a loan in
##     the categorical variable loan_default, where 1=loan received and 0=no loan given.
##     From the various attrbutes of each person, the model predicts if a person receives
##     a loan, classifying them. The accuracy of the model is very low at 0.595, due to
##     our balancing of the original dataset.
```

```
#(2) Logit Model
Loan_logit <- glm(loan_default ~ ., data = train, family = "binomial")
summary(Loan_logit)
```

```
##
## Call:
## glm(formula = loan_default ~ ., family = "binomial", data = train)
##
## Coefficients:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)               -1.357e+00  2.182e-01  -6.220 4.96e-10 ***
## loan_amnt                 -3.679e-05  1.061e-05  -3.468 0.000524 ***
## adjusted_annual_inc        1.100e-06  1.684e-06   0.653 0.513674
## pct_loan_income            3.351e+00  7.165e-01   4.677 2.91e-06 ***
## dti                        8.959e-03  5.713e-03   1.568 0.116845
## residence_propertyRent     4.030e-01  7.860e-02   5.127 2.94e-07 ***
## months_since_first_credit -4.674e-04  4.622e-04  -1.011 0.311855
## inq_last_6mths             2.671e-01  3.819e-02   6.995 2.65e-12 ***
## open_acc                   1.277e-02  9.339e-03   1.367 0.171522
## bc_util                    6.093e-03  1.538e-03   3.962 7.42e-05 ***
## num_accts_ever_120_pd      2.077e-02  4.249e-02   0.489 0.624944
## pub_rec_bankruptcies       7.096e-02  1.250e-01   0.568 0.570161
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4158.8  on 2999  degrees of freedom
## Residual deviance: 4004.6  on 2988  degrees of freedom
## AIC: 4028.6
##
## Number of Fisher Scoring iterations: 4
```

```r
#Predictions of logit model and threshold of .5
logit_probabilities <- predict(Loan_logit, newdata = test, type = "response")
logit_preds <- ifelse(logit_probabilities > 0.5, 1, 0)

#Confusion matrix
logit_cm <- table(Predicted = logit_preds, Actual = test$loan_default)
logit_cm
```

```
##          Actual
## Predicted   0   1
##         0 304 194
##         1 202 300
```

```r
confusionMatrix(logit_cm)
```

```
## Confusion Matrix and Statistics
##
##          Actual
## Predicted   0   1
##         0 304 194
##         1 202 300
##
##                Accuracy : 0.604
##                  95% CI : (0.5729, 0.6345)
##     No Information Rate : 0.506
##     P-Value [Acc > NIR] : 2.99e-10
##
##                   Kappa : 0.208
##
##  Mcnemar’s Test P-Value : 0.725
##
##             Sensitivity : 0.6008
##             Specificity : 0.6073
##          Pos Pred Value : 0.6104
##          Neg Pred Value : 0.5976
##              Prevalence : 0.5060
##          Detection Rate : 0.3040
##    Detection Prevalence : 0.4980
##       Balanced Accuracy : 0.6040
##
##        ’Positive’ Class : 0
##
```

```r
cat("300 True positives
    304 True negatives
    202 False positives
    194 False negatives
    The accuracy is quite low at 0.604, because we balanced the original dataset.")
```

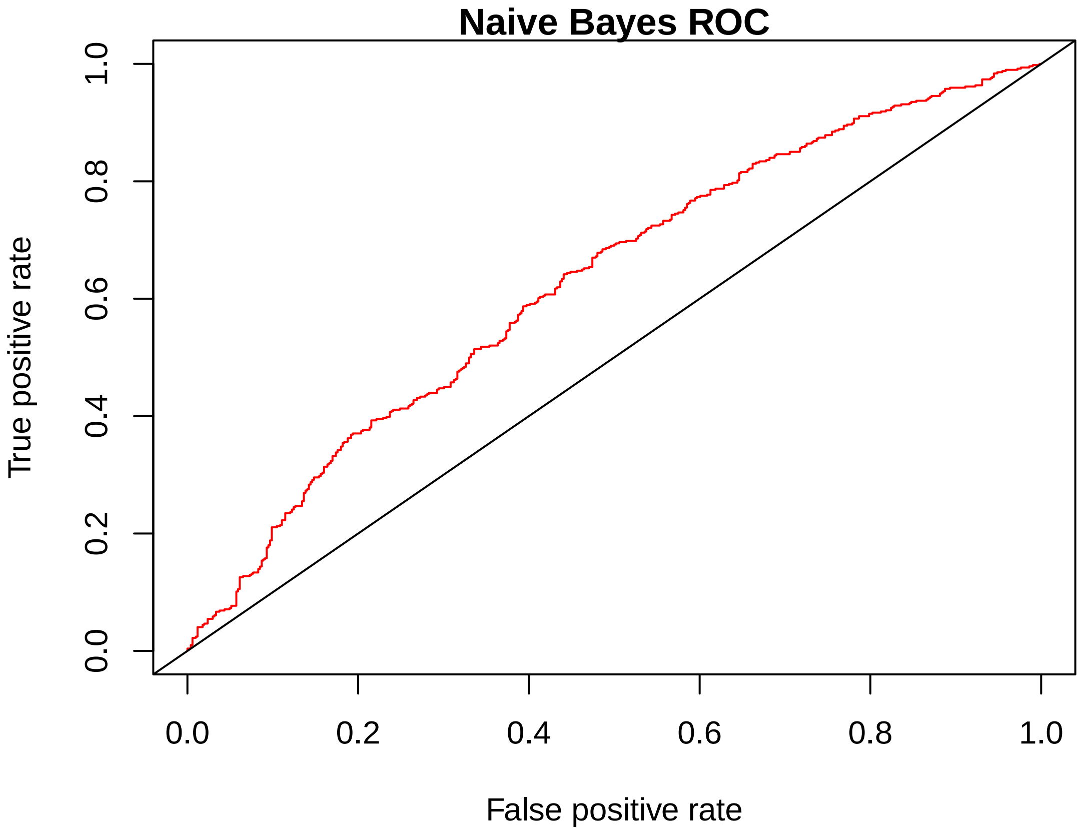
```
## 300 True positives
##     304 True negatives
##     202 False positives
##     194 False negatives
##     The accuracy is quite low at 0.604, because we balanced the original dataset.
```

```
#(3) Model accuracy comparison and ROC curves

cat("The Naive Bayes model yielded an accuracy of 0.595 while the logistic regression
    model yielded an accuracy of 0.604. The logit model is marginally more accurate
    than the NB model, therefore we can say that the logit model is a better fit
    for predicting whether or not a person accurately receives a loan from loan_default.
    To further analyze the models we must compute the ROC curve and AuC from the ROCs.")
```

```
## The Naive Bayes model yielded an accuracy of 0.595 while the logistic regression
##     model yielded an accuracy of 0.604. The logit model is marginally more accurate
##     than the NB model, therefore we can say that the logit model is a better fit
##     for predicting whether or not a person accurately receives a loan from loan_default.
##     To further analyze the models we must compute the ROC curve and AuC from the ROCs.
```

```
#Naive Bayes ROC and AuC
NB_prob <- predict(NB, newdata = test, type = "raw")
NB_pred <- prediction(NB_prob[,2], test$loan_default)
NB_perf <- performance(NB_pred, measure = "tpr", x.measure = "fpr")
#following order: bottom, left, top, and right.
par(mar=c(5,8,1,.5))
#Receiver operating characteristic
plot(NB_perf, col="red",main="Naive Bayes ROC")
abline(a=0, b=1)
```



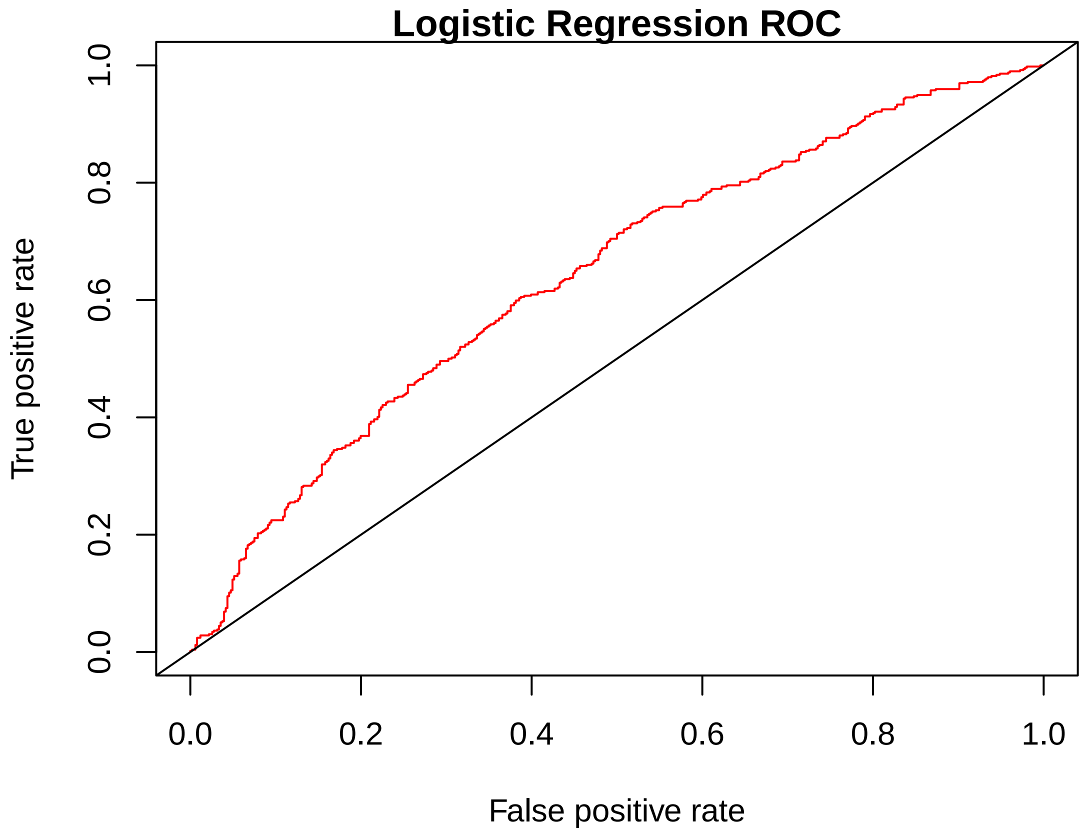
```r
auc <- performance(NB_pred, measure = "auc") #Calculates AuC
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.6280704
```

```r
cat("The ROC curve for the Naive Bayes model is not very strong, since it is closer
    to the reference line than the top left of the plot, and we observe from the
    calculated AuC value of 0.6280704, that ~62.81% of the time the model accurately
    distinguishes a person with positives traits to having a loan_default=1 over a person
    with negative traits. A good AuC value for a model is, .8, and .5 means the model is
    compeltely random, therefore we can conclude that the NB model has very low efficacy
    and is a bad/weak classifier for loan_default.")
```

```
## The ROC curve for the Naive Bayes model is not very strong, since it is closer
##     to the reference line than the top left of the plot, and we observe from the
##     calculated AuC value of 0.6280704, that ~62.81% of the time the model accurately
##     distinguishes a person with positives traits to having a loan_default=1 over a person
##     with negative traits. A good AuC value for a model is, .8, and .5 means the model is
##     compeltely random, therefore we can conclude that the NB model has very low efficacy
##     and is a bad/weak classifier for loan_default.
```

```r
#Logistic regression ROC and AuC
logit_prob <- predict(Loan_logit, newdata = test, type = "response")
logit_pred <- prediction(logit_prob, test$loan_default)
logit_perf <- performance(logit_pred, measure = "tpr", x.measure = "fpr")
#following order: bottom, left, top, and right.
par(mar=c(5,8,1,.5))
#Receiver operating characteristic
plot(logit_perf, col="red", main="Logistic Regression ROC")
abline(a=0, b=1)
```



```r
auc <- performance(logit_pred, measure = "auc") #Calculates AuC
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.6412003
```

```r
cat("The ROC curve for the Naive Bayes model is not very strong, since it is closer
    to the reference line than the top left of the plot, and we observe from the
    calculated AuC value of 0.6412003, that ~64.12% of the time the model accurately
    distinguishes a person with positives traits to having a loan_default=1 over a person
    with negative traits. A good AuC value for a model is, .8, and .5 means the model is
    compeltely random, therefore we can conclude that the NB model has very low efficacy
    and is a bad/weak classifier for loan_default.")
```

```
## The ROC curve for the Naive Bayes model is not very strong, since it is closer
##     to the reference line than the top left of the plot, and we observe from the
##     calculated AuC value of 0.6412003, that ~64.12% of the time the model accurately
##     distinguishes a person with positives traits to having a loan_default=1 over a person
##     with negative traits. A good AuC value for a model is, .8, and .5 means the model is
##     compeltely random, therefore we can conclude that the NB model has very low efficacy
##     and is a bad/weak classifier for loan_default.
```

```r
cat("While both models are very bad classifiers for loan_default, the ROC curves
    for both models are very similar, but the AuC for the logit model is a bit
    higher than the Naive Bayes model, at 0.6412003 > 0.6280704. Therefore we can
    say that the logit model has a higher efficacy and likelihood to ranking
    positive instsances higher than negative instances compared to the NB model.
    Where ranking details which people receive loans and which people do not, based
    on their positive or negative predictors.")
```

```
## While both models are very bad classifiers for loan_default, the ROC curves
##     for both models are very similar, but the AuC for the logit model is a bit
##     higher than the Naive Bayes model, at 0.6412003 > 0.6280704. Therefore we can
##     say that the logit model has a higher efficacy and likelihood to ranking
##     positive instsances higher than negative instances compared to the NB model.
##     Where ranking details which people receive loans and which people do not, based
##     on their positive or negative predictors.
```