Logistic Regression Exercises

Ethan Marcano

2/25/2022

Exercise 1: Predicting the Baseball World Series Champion

A) Exploring the dataset

```
head(baseball)
```

i) Each row of the dataset represent's a playoff's team's performance in a particular year. Through the years, different numbers of teams have been invited to the playoffs. How has the number of teams making it to the playoffs changed, according to this dataset?

```
##   Team League Year  RS  RA   W   OBP   SLG    BA RankSeason RankPlayoffs
## 1  NYY     AL 1962 817 680  96 0.337 0.426 0.267          2            1
## 2  SFG     NL 1962 878 690 103 0.341 0.441 0.278          1            2
## 3  LAD     NL 1963 640 550  99 0.309 0.357 0.251          2            1
## 4  NYY     AL 1963 714 547 104 0.309 0.403 0.252          1            2
## 5  NYY     AL 1964 730 577  99 0.317 0.387 0.253          1            2
## 6  STL     NL 1964 715 652  93 0.324 0.392 0.272          2            1
##   NumCompetitors WonWorldSeries
## 1              2              1
## 2              2              0
## 3              2              1
## 4              2              0
## 5              2              0
## 6              2              1
```

```
tail(baseball)
```

```
##     Team League Year  RS  RA  W   OBP   SLG    BA RankSeason RankPlayoffs
```

```
## 239  NYY     AL 2012 804 668 95 0.337 0.453 0.265          3            3
## 240  OAK     AL 2012 713 614 94 0.310 0.404 0.238          4            4
## 241  SFG     NL 2012 718 649 94 0.327 0.397 0.269          4            1
## 242  STL     NL 2012 765 648 88 0.338 0.421 0.271          6            3
## 243  TEX     AL 2012 808 707 93 0.334 0.446 0.273          5            5
## 244  WSN     NL 2012 731 594 98 0.322 0.428 0.261          1            4
##     NumCompetitors WonWorldSeries
## 239             10              0
## 240             10              0
## 241             10              1
## 242             10              0
## 243             10              0
## 244             10              0
```

```
summary(baseball)
```

```
##      Team              League               Year            RS
##  Length:244         Length:244         Min.   :1962   Min.   : 583.0
##  Class :character   Class :character   1st Qu.:1982   1st Qu.: 730.0
##  Mode  :character   Mode  :character   Median :1998   Median : 780.5
##                                        Mean   :1993   Mean   : 786.3
##                                        3rd Qu.:2005   3rd Qu.: 836.0
##                                        Max.   :2012   Max.   :1009.0
##        RA              W               OBP              SLG
##  Min.   :472.0   Min.   : 82.00   Min.   :0.2980   Min.   :0.3350
##  1st Qu.:614.0   1st Qu.: 91.00   1st Qu.:0.3280   1st Qu.:0.3990
##  Median :661.5   Median : 95.00   Median :0.3380   Median :0.4200
##  Mean   :666.1   Mean   : 95.12   Mean   :0.3373   Mean   :0.4191
##  3rd Qu.:711.0   3rd Qu.: 98.00   3rd Qu.:0.3460   3rd Qu.:0.4373
##  Max.   :903.0   Max.   :116.00   Max.   :0.3730   Max.   :0.4910
##        BA           RankSeason     RankPlayoffs   NumCompetitors
##  Min.   :0.2350   Min.   :1.000   Min.   :1.000   Min.   : 2.00
##  1st Qu.:0.2597   1st Qu.:2.000   1st Qu.:2.000   1st Qu.: 4.00
##  Median :0.2670   Median :3.000   Median :3.000   Median : 8.00
##  Mean   :0.2668   Mean   :3.123   Mean   :2.717   Mean   : 6.23
##  3rd Qu.:0.2740   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.: 8.00
##  Max.   :0.2930   Max.   :8.000   Max.   :5.000   Max.   :10.00
```

```
##  WonWorldSeries
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1926
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

```
filterball <- baseball %>%
    select(Team, Year, RankSeason, NumCompetitors, WonWorldSeries) %>%
    filter(WonWorldSeries == 1)


head(filterball)
```

```
##   Team Year RankSeason NumCompetitors WonWorldSeries
## 1  NYY 1962          2              2              1
## 2  LAD 1963          2              2              1
## 3  STL 1964          2              2              1
## 4  LAD 1965          2              2              1
## 5  BAL 1966          1              2              1
## 6  STL 1967          1              2              1
```

- The number of competitors has increased from two teams at the beginning to 10 teams in the present day.
- Teams invited to the playoffs are somewhat lower in the regular season because of the the number of competitors invited.

ii) Given that a team has made it to the playoffs, it is much harder to win the World Series if there are 10 teams competing for the championship versus just two. Therefore we have the variable NumCompetitors in our dataset. NumCompetitors contains the number of total teams making the playoffs in the year of the observation. For instance, NumCompetitors is 2 for the 1962 New York Yankees, but it is 8 for the 1998 Boston Red Sox. Without knowing anything else about the teams in the playoffs, can you think of a simple model that uses NumCompetitors to predict the probability of a team winning?

- A linear regression model using NumCompetitors shows that it is statistically significant in determining the probability of a team winning.

```
simplemodel <- lm(WonWorldSeries ~ NumCompetitors, data = dfbaseball)
```

B) Building a logistic regression model to predict the winner

i) When we are not sure which of our variables are useful in predicting a particular outcome, it is often helpful to build bivariate models, which are models that predict the outcome using a single independent variable. Build a bivariate logistic regression model using each of the following variables as the independent variable to predict WonWorldSeries and the entire dataset as the training set each time: Year, RS, RA, W, OBP, SLG, BA, RankSeason, NumCompetitors, and League. You should have created 10 logistic regression models. Describe each of the models by giving the regression equation and the accuracy of the model. For which models is the independent variable significant? In your opinion, which are the best models and why?

- $$log(Odds) = 72.236 + -0.037(Year)$$
- $$log(Odds) = 0.66 + -0.0026(RS)$$
- $$log(Odds) = 1.88 + -0.005(RA)$$
- $$log(Odds) = -6.856 + -0.0567(W)$$
- $$log(Odds) = 2.741 + -12.4(OBP)$$
- $$log(Odds) = 3.2 + -11.13(SLG)$$
- $$log(Odds) = -0.64 + -2.98(BA)$$
- $$log(Odds) = -0.825 + -0.2(RankSeason)$$
- $$log(Odds) = .0387 + -.252(NumCompetitors)$$
- $$log(Odds) = -1.35 + -.1583(LeagueNL)$$

- The year, number of competitors, RankSeason, and RA are all significant in their individual models.

ii) Now, build a logistic model using all of the variables that you found to be significant in the bivariate models as the independent variables, and the entire dataset to train the model. Are all of the independent variables significant in this model? Why would some independent variables be significant in the bivariate model using that variable, but then not significant in a model that uses more than one independent variables? Be sure to provide numerical evidence for your claim.

- As none of the p-values are significant, none of the independent variables that were originally found to be significant are significant in the multivariate model. This may be due to colinearity.

```
multimodel <- glm(WonWorldSeries ~ NumCompetitors + Year + RA +
    RankSeason, family = "binomial", data = dfbaseball)
summary(multimodel)
```

```
##
## Call:
## glm(formula = WonWorldSeries ~ NumCompetitors + Year + RA + RankSeason,
##     family = "binomial", data = dfbaseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0336  -0.7689  -0.5139  -0.4583   2.2195
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    12.5874376 53.6474210   0.235    0.814
## NumCompetitors -0.1794264  0.1815933  -0.988    0.323
## Year           -0.0061425  0.0274665  -0.224    0.823
## RA             -0.0008238  0.0027391  -0.301    0.764
## RankSeason     -0.0685046  0.1203459  -0.569    0.569
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 226.37  on 239  degrees of freedom
## AIC: 236.37
##
```

```
## Number of Fisher Scoring iterations: 4
```

```
bestmodel <- glm(WonWorldSeries ~ NumCompetitors + Year, family = "binomial",
    data = dfbaseball)
summary(bestmodel)
```

iii) Using any number of the independent variables that you found to be significant in the bivariate models, find what you think is the best model, and justify why you think it is the best. How many independent variables are used in your final model?

```
##
## Call:
## glm(formula = WonWorldSeries ~ NumCompetitors + Year, family = "binomial",
##     data = dfbaseball)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0050  -0.7823  -0.5115  -0.4970   2.2552
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)    13.350467  53.481896   0.250    0.803
## NumCompetitors -0.212610   0.175520  -1.211    0.226
## Year           -0.006802   0.027328  -0.249    0.803
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 239.12  on 243  degrees of freedom
## Residual deviance: 226.90  on 241  degrees of freedom
## AIC: 232.9
##
## Number of Fisher Scoring iterations: 4
```

- I used the independent variables with the most significance on their own (NumCompetitors and Year) to see if it is the best model. It has a lower AIC than the other multivariate model, and while the p-values are still high, are lower than the other model.

iv) Do your findings in this problem confirm or reject the claim that the playoffs is more about luck than skill? Why?

- Yes, it confirms that the claim. The most significant variable, NumCompetitors, has no bearing on a team's skill.

Exercise 2: Predicting Parole Violators

```
parole <- read.csv("Parole.csv")
dfparole <- as_tibble(parole)
```

```
summary(dfparole)
```

```
##       Male          RaceWhite           Age            State
##  Min.   :0.0000   Min.   :0.0000   Min.   :18.40   Length:675
##  1st Qu.:1.0000   1st Qu.:0.0000   1st Qu.:25.35   Class :character
##  Median :1.0000   Median :1.0000   Median :33.70   Mode  :character
##  Mean   :0.8074   Mean   :0.5763   Mean   :34.51
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:42.55
##  Max.   :1.0000   Max.   :1.0000   Max.   :67.00
##    TimeServed     MaxSentence    MultipleOffenses    Crime
##  Min.   :0.000   Min.   : 1.00   Min.   :0.0000   Length:675
##  1st Qu.:3.250   1st Qu.:12.00   1st Qu.:0.0000   Class :character
##  Median :4.400   Median :12.00   Median :1.0000   Mode  :character
##  Mean   :4.198   Mean   :13.06   Mean   :0.5363
##  3rd Qu.:5.200   3rd Qu.:15.00   3rd Qu.:1.0000
##  Max.   :6.000   Max.   :18.00   Max.   :1.0000
##     Violator
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.1156
##  3rd Qu.:0.0000
##  Max.   :1.0000
```

A) How many parolees do we have data for? Of the parolees that we have data for, what percentage violated the terms of their parole?

```
count(dfparole)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1   675
```

```
violators <- dfparole %>%
    select(everything()) %>%
    filter(Violator == 1)


count(violators)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    78
```

- We have 675 parolees. Out of that 675, 78 of them violated parole.

B) Randomly split the data into a training set and a testing set, putting 70% of the data in the training set. Then, build a logistic regression model to predict the variable Violator using all of the other variables as independent variables. You should use the training dataset to build the model.

```
dfparole$State = as.factor(dfparole$State)
dfparole$Crime = as.factor(dfparole$Crime)
summary(dfparole$State)
```

```
##  Kentucky Louisiana     Other  Virginia
##       120        82       143       330
```

```
summary(dfparole$Crime)
```

```
## Driving   Drugs Larceny   Other
##     101     153     106     315
```

```
set.seed(88)
spl = sample.split(dfparole$Violator, SplitRatio = 0.7)


paroletrain = subset(dfparole, spl == TRUE)
paroletest = subset(dfparole, spl == FALSE)


nrow(paroletrain)/nrow(dfparole)
```

```
## [1] 0.7007407
```

```
nrow(paroletest)/nrow(dfparole)
```

```
## [1] 0.2992593
```

```
trainmodel <- glm(Violator ~ ., family = binomial, data = paroletrain)
```

```
summary(trainmodel)
```

i) Describe your resulting model. Which variables are significant in your model?

```
##
## Call:
## glm(formula = Violator ~ ., family = binomial, data = paroletrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7221  -0.3959  -0.2403  -0.1494   2.8212
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -3.29370    1.47788  -2.229 0.025836 *
## Male              0.65662    0.47189   1.391 0.164082
## RaceWhite        -0.67930    0.42425  -1.601 0.109338
## Age               0.01739    0.01662   1.046 0.295452
## StateLouisiana    0.67688    0.60992   1.110 0.267087
## StateOther       -0.17308    0.54082  -0.320 0.748949
## StateVirginia    -3.38536    0.73642  -4.597 4.28e-06 ***
## TimeServed       -0.06809    0.11415  -0.596 0.550863
## MaxSentence       0.04536    0.05227   0.868 0.385552
## MultipleOffenses  1.42426    0.39268   3.627 0.000287 ***
## CrimeDrugs       -0.23931    0.67429  -0.355 0.722655
## CrimeLarceny      0.99710    0.69991   1.425 0.154266
## CrimeOther        0.19106    0.58920   0.324 0.745731
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 340.04  on 472  degrees of freedom
## Residual deviance: 242.18  on 460  degrees of freedom
```

```
## AIC: 268.18
##
## Number of Fisher Scoring iterations: 6
```

- Some significant variables are State$Virginia and MultipleOffenses.

```
Male = 1
RaceWhite = 1
Age = 50
StateOther = 1
StateLouisiana = 0
StateVirginia = 0
time.served = 3
max.sentence = 12
multiple.offenses = 0
CrimeDrugs = 0
CrimeLarceny = 1
CrimeOther = 0

logodds = -3.2937 + 0.65662 * Male + -0.6793 * RaceWhite + 0.01739 *
    Age + 0.67688 * StateOther + -0.17308 * StateLouisiana +
    -3.38536 * StateVirginia + -0.06809 * time.served + 0.04536 *
    max.sentence + 1.42426 * multiple.offenses + -0.23931 * CrimeDrugs +
    0.9971 * CrimeLarceny + 0.19106 * CrimeOther

logodds
```

ii) Consider a parolee who is male, of white race, aged 50 years at prison release, from the state of Maryland, served 3 months, had a maximum sentence of 12 months, did not commit multiple offenses, and committed a larceny. According to your model, what is the probability that this individual is a violator?

```
## [1] -0.43285
```

```
odds = exp(logodds)
odds
```

```
## [1] 0.6486578
```

```
1/(1 + exp(-logodds))
```

```
## [1] 0.393446
```

- There is a 39% chance that the Parolee is a violator.

```
predictions = predict(trainmodel, newdata = paroletest, type = "response")


summary(predictions)
```

iii) Now compute the model's predicted probabilities for parolees in the testing set. Then create a confusion matrix for the test set using a threshold of 0.5. What is the model's false positive rate on the test set? False negative rate? Overall accuracy?

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001999 0.018725 0.044731 0.091047 0.102107 0.823355
```

```
testmatrix <- table(paroletest$Violator, as.numeric(predictions >=
    0.5))
testmatrix
```

```
##
##       0   1
##   0 179   0
##   1  20   3
```

```
sum(diag(testmatrix))/sum(testmatrix)
```

```
## [1] 0.9009901
```

- Maximum probability is 82%. There are 20 False Negatives and 0 False Positives. The overall accuracy for the model is 90.1%

```
simpletest <- table(paroletest$Violator)
simpletest
```

iv) Compare your accuracy on the test set to a baseline model that predicts every parolee in the test set is a non-violator, regardless of the values of the independent variables. Does your model improve over this simple model?

```
##
##   0   1
```

```
## 179  23
```

```
simpletest[1]/simpletest[2]
```

```
##        0
## 7.782609
```

- Our model does improve over the simple model.

```
highthreshold <- table(paroletest$Violator, as.numeric(predictions >=
    0.7))

lowthreshold <- table(paroletest$Violator, as.numeric(predictions >=
    0.3))

highthreshold
```

v) Consider a parole board using the model to predict whether parolees will be violators or not. The job of a parole board is to make sure that a prisoners is ready to be released into free society, and therefore parole boards tend to be particularly concerned about releasing prisoners who will violate their parole. Would the parole board be more concerned by false positive errors or false negative errors? How should they adjust their threshold to reflect their error preferences?

```
##
##       0   1
##   0 179   0
##   1  22   1
```

```
lowthreshold
```

```
##
##       0   1
##   0 173   6
##   1  15   8
```

- The board would be more concerned with a false negative, being that means that a parolee has violated parole and committed another crime. A false positive, wherein a prisoner is denied parole would induce less regret in the parole board. We would want more false positives as opposed to false negatives, and we would adjust the threshold to be lower to reflect that.

```
predrocr <- prediction(predictions, paroletest$Violator)


as.numeric(performance(predrocr, "auc")@y.values)
```

vi) Compute the AUC of the model on the test set, and interpret what the number means in this context. Considering the AUC, the accuracy compared to the base model, and what happens when the threshold is adjusted, do you think this model is of value to a parole board? Why or why not?

```
## [1] 0.8214719
```

- AUC = 0.8214719. I think that considering the different measures of accuracy, that the model is still of value to a parole board, especially in a context where it is better to be safe than sorry.

C) How can we improve our dataset to best address selection bias?

- It would help to include the missing prisoners and labeling them as non-violators since it is technically true. It would be better if we had the true outcome of different parolees, but that may require a larger dataset.

Exercise 3: Loan Repayment

A) Building a logistic regression model

```
loans <- read.csv("Loans.csv")
dfloans <- as_tibble(loans)

summary(dfloans)
```

i) Randomly split the dataset into a training set and a testing set. Put 70% of the data in the training set. What is the accuracy on the test set of a simple baseline model that predicts that all loans will be paid back in full(NotFullyPaid = 0)? Our goal will be to build a model that dds value over this simple baseline method.

```
##   CreditPolicy     Purpose             IntRate        Installment
##  Min.   :0.000   Length:9578        Min.   :0.0600   Min.   : 15.67
##  1st Qu.:1.000   Class :character   1st Qu.:0.1039   1st Qu.:163.77
##  Median :1.000   Mode  :character   Median :0.1221   Median :268.95
##  Mean   :0.805                      Mean   :0.1226   Mean   :319.09
##  3rd Qu.:1.000                      3rd Qu.:0.1407   3rd Qu.:432.76
##  Max.   :1.000                      Max.   :0.2164   Max.   :940.14
##   LogAnnualInc          Dti              Fico       DaysWithCrLine
##  Min.   : 7.548   Min.   : 0.000   Min.   :612.0   Min.   :  179
##  1st Qu.:10.558   1st Qu.: 7.213   1st Qu.:682.0   1st Qu.: 2820
##  Median :10.929   Median :12.665   Median :707.0   Median : 4140
##  Mean   :10.932   Mean   :12.607   Mean   :710.8   Mean   : 4561
##  3rd Qu.:11.291   3rd Qu.:17.950   3rd Qu.:737.0   3rd Qu.: 5730
##  Max.   :14.528   Max.   :29.960   Max.   :827.0   Max.   :17640
##     RevolBal          RevolUtil      InqLast6mths      Delinq2yrs
##  Min.   :      0   Min.   :  0.0   Min.   : 0.000   Min.   : 0.0000
##  1st Qu.:   3187   1st Qu.: 22.6   1st Qu.: 0.000   1st Qu.: 0.0000
##  Median :   8596   Median : 46.3   Median : 1.000   Median : 0.0000
##  Mean   :  16914   Mean   : 46.8   Mean   : 1.577   Mean   : 0.1637
##  3rd Qu.:  18250   3rd Qu.: 70.9   3rd Qu.: 2.000   3rd Qu.: 0.0000
##  Max.   :1207359   Max.   :119.0   Max.   :33.000   Max.   :13.0000
##      PubRec         NotFullyPaid
```

```
##  Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.0000
##  Mean   :0.06212   Mean   :0.1601
##  3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :5.00000   Max.   :1.0000
```

```
filterloans <- dfloans %>%
    select(everything()) %>%
    filter(NotFullyPaid == 0)


count(filterloans)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1  8045
```

```
set.seed(88)
spl = sample.split(dfloans$NotFullyPaid, SplitRatio = 0.7)


loanstrain = subset(dfloans, spl == TRUE)
loanstest = subset(dfloans, spl == FALSE)


nrow(loanstrain)/nrow(dfloans)
```

```
## [1] 0.7000418
```

```
nrow(loanstest)/nrow(dfloans)
```

```
## [1] 0.2999582
```

```
simpleloan <- table(loanstest$NotFullyPaid)
simpleloan
```

```
##
##    0    1
## 2413  460
```

```
simpleloan[1]/sum(simpleloan)
```

```
##         0
## 0.8398886
```

- The baseline model is 83.99% accurate.

```
loanmodel = glm(NotFullyPaid ~ ., data = loanstrain, family = "binomial")

summary(loanmodel)
```

ii) Build a logistic regression model that predicts the dependent variable NotFullyPaid using all of the other variables as independent variables. Use the training set as the data for the model. Describe your resulting model. Which of the independent variables are significant in your model?

```
##
## Call:
## glm(formula = NotFullyPaid ~ ., family = "binomial", data = loanstrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.9519  -0.6151  -0.4933  -0.3586   2.5173
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 9.363e+00  1.574e+00   5.949 2.69e-09 ***
## CreditPolicy               -3.183e-01  1.007e-01  -3.162 0.001565 **
## Purposecredit_card         -4.752e-01  1.298e-01  -3.661 0.000251 ***
## Purposedebt_consolidation -3.243e-01  9.319e-02  -3.480 0.000501 ***
## Purposeeducational          1.030e-01  1.796e-01   0.574 0.566265
## Purposehome_improvement     4.873e-02  1.551e-01   0.314 0.753426
## Purposemajor_purchase      -2.536e-01  1.982e-01  -1.280 0.200622
## Purposesmall_business       5.993e-01  1.402e-01   4.274 1.92e-05 ***
## IntRate                     6.443e-01  2.114e+00   0.305 0.760486
## Installment                 1.311e-03  2.113e-04   6.207 5.41e-10 ***
## LogAnnualInc               -4.586e-01  7.359e-02  -6.232 4.60e-10 ***
## Dti                        -7.458e-03  5.532e-03  -1.348 0.177634
## Fico                       -9.211e-03  1.723e-03  -5.346 9.01e-08 ***
## DaysWithCrLine             -1.954e-06  1.629e-05  -0.120 0.904507
## RevolBal                    5.082e-06  1.170e-06   4.342 1.41e-05 ***
## RevolUtil                   3.564e-03  1.532e-03   2.326 0.020002 *
## InqLast6mths                8.972e-02  1.647e-02   5.447 5.13e-08 ***
```

```
## Delinq2yrs                -3.160e-02  6.803e-02  -0.465 0.642249
## PubRec                     2.878e-01  1.152e-01   2.498 0.012498 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5896.6  on 6704  degrees of freedom
## Residual deviance: 5467.9  on 6686  degrees of freedom
## AIC: 5505.9
##
## Number of Fisher Scoring iterations: 5
```

- A surprising amount of independent variables are strongly significant. The significant independent variables are Purposecredit_card, Purposedebt_consolidation, Purposesmall_business, Installment, LogAnnualInc, Fico, RevolBal, and InqLast6mths.

```
# fico coefficient is -9.211e-03 and the score difference
# is 10

logoddiff = -0.009211 * -10
logoddiff
```

iii) Consider two loan applications, which are identical other than the fact that the borrower in App. A has a FICO score of 700 while the borrow in App. B has a FICO score of 710. Let Logit(A) be the function of loan A not being paid back in full (according to our model) and define Logit(B) similarly. What is the value of Logit(A) - Logit(B)?

```
## [1] 0.09211
```

- Since the two applications are the same except for the difference in FICO score, the predicted logodds of A differ by .09211 from B.

```
loanstest$PredictedRisk <- predict(loanmodel, newdata = loanstest,
    type = "response")

loanmatrix = table(loanstest$NotFullyPaid, loanstest$PredictedRisk >
```

```
    0.5)

loanmatrix
```

iv) Now predict the probability of the test set loans not being paid back in full. Store these predicted probabilities in a variable named PredictedRisk and add it to your test set. What is the accuracy of the logisitic regression model on the test set using a threshold of 0.5? How does this compare to baseline?

```
##
##     FALSE TRUE
##   0  2394   19
##   1   447   13
```

```
sum(diag(loanmatrix))/sum(loanmatrix)
```

```
## [1] 0.8378002
```

- The accuracy of the logistic regression model is slightly worse, if not the same as the baseline model.

```
pred = prediction(loanstest$PredictedRisk, loanstest$NotFullyPaid)
as.numeric(performance(pred, "auc")@y.values)
```

v) What is the test set AUC of the model? Given the accuracy and the AUC, would this model be useful to an investor?

```
## [1] 0.6673868
```

- The AUC is 0.6674. This model is only somewhat better than a coinflip, so it is unlikely to be useful to an investor.

B) Using the loan's interest rate as a "smart baseline" to order the loans according to risk.

```
interestmodel <- glm(NotFullyPaid ~ IntRate, data = loanstrain,
    family = binomial)

summary(interestmodel)
```

i) Build a logistic regression model that predicts the dependent variable NotFullyPaid using IntRate as the only independent variable. Was it significant in the first model you built? How would you explain the difference?

```
##
## Call:
```

```
## glm(formula = NotFullyPaid ~ IntRate, family = binomial, data = loanstrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0751  -0.6291  -0.5412  -0.4324   2.3020
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -3.7369     0.1709  -21.87   <2e-16 ***
## IntRate      16.4614     1.2887   12.77   <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: 5896.6  on 6704  degrees of freedom
## Residual deviance: 5728.1  on 6703  degrees of freedom
## AIC: 5732.1
##
## Number of Fisher Scoring iterations: 4
```

- IntRate is very significant in this model. It did not have significance in the original model. I would explain the difference via correlation.

```
dfloans$Purpose = as.numeric(dfloans$Purpose)
```

```
## Warning: NAs introduced by coercion
```

```
summary(dfloans$Purpose)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##      NA      NA      NA     NaN      NA      NA    9578
```

```
cor(dfloans)
```

```
##                CreditPolicy Purpose     IntRate  Installment LogAnnualInc
## CreditPolicy     1.00000000      NA -0.29408909  0.058769616   0.03490601
## Purpose                  NA       1          NA           NA           NA
## IntRate         -0.29408909      NA  1.00000000  0.276140176   0.05638254
## Installment      0.05876962      NA  0.27614018  1.000000000   0.44810215
## LogAnnualInc     0.03490601      NA  0.05638254  0.448102154   1.00000000
```

```
## Dti             -0.09090057      NA  0.22000563  0.050201841  -0.05406476
## Fico             0.34831868      NA -0.71482077  0.086039394   0.11457595
## DaysWithCrLine   0.09902619      NA -0.12402216  0.183297427   0.33689639
## RevolBal        -0.18751848      NA  0.09252705  0.233625400   0.37213960
## RevolUtil       -0.10409495      NA  0.46483728  0.081356217   0.05488106
## InqLast6mths    -0.53551118      NA  0.20278026 -0.010418675   0.02917129
## Delinq2yrs      -0.07631843      NA  0.15607873 -0.004367654   0.02920327
## PubRec          -0.05424305      NA  0.09816221 -0.032759675   0.01650648
## NotFullyPaid    -0.15811915      NA  0.15955158  0.049955162  -0.03343938
##                         Dti        Fico DaysWithCrLine    RevolBal   RevolUtil
## CreditPolicy   -0.090900569  0.34831868     0.09902619 -0.18751848 -0.10409495
## Purpose                  NA          NA             NA          NA          NA
## IntRate         0.220005629 -0.71482077    -0.12402216  0.09252705  0.46483728
## Installment     0.050201841  0.08603939     0.18329743  0.23362540  0.08135622
## LogAnnualInc   -0.054064762  0.11457595     0.33689639  0.37213960  0.05488106
## Dti             1.000000000 -0.24119099     0.06010112  0.18874778  0.33710918
## Fico           -0.241190985  1.00000000     0.26387975 -0.01555250 -0.54128934
## DaysWithCrLine  0.060101120  0.26387975     1.00000000  0.22934416 -0.02423925
## RevolBal        0.188747784 -0.01555250     0.22934416  1.00000000  0.20377904
## RevolUtil       0.337109179 -0.54128934    -0.02423925  0.20377904  1.00000000
## InqLast6mths    0.029189016 -0.18529299    -0.04173642  0.02239448 -0.01387989
## Delinq2yrs     -0.021792180 -0.21633953     0.08137375 -0.03324306 -0.04273999
## PubRec          0.006208759 -0.14759196     0.07182617 -0.03100964  0.06671655
## NotFullyPaid    0.037361524 -0.14966630    -0.02923667  0.05369936  0.08208777
##                InqLast6mths   Delinq2yrs       PubRec NotFullyPaid
## CreditPolicy    -0.53551118 -0.076318433 -0.054243047 -0.158119150
## Purpose                  NA           NA           NA           NA
## IntRate          0.20278026  0.156078730  0.098162208  0.159551583
## Installment     -0.01041868 -0.004367654 -0.032759675  0.049955162
## LogAnnualInc     0.02917129  0.029203269  0.016506475 -0.033439377
## Dti              0.02918902 -0.021792180  0.006208759  0.037361524
## Fico            -0.18529299 -0.216339530 -0.147591956 -0.149666303
## DaysWithCrLine  -0.04173642  0.081373752  0.071826169 -0.029236672
## RevolBal         0.02239448 -0.033243065 -0.031009638  0.053699363
## RevolUtil       -0.01387989 -0.042739992  0.066716548  0.082087768
## InqLast6mths     1.00000000  0.021245402  0.072672891  0.149451944
```

```
## Delinq2yrs       0.02124540  1.000000000  0.009184189  0.008881041
## PubRec           0.07267289  0.009184189  1.000000000  0.048634301
## NotFullyPaid     0.14945194  0.008881041  0.048634301  1.000000000
```

```
interestpred = predict(interestmodel, newdata = loanstest, type = "response")


summary(interestpred)
```

ii) Use the new model to make predictions for the observations in the test set. What is the highest predicted probability of a loan being paid in full? How many loans would we predict would not be paid back in full if we used a threshold of 0.5 to make predictions?

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.06013 0.11459 0.15267 0.16225 0.19557 0.45644
```

- The highest probability is 0.4564. If we used a 0.5 threshold, it would mean that no loans would be predicted as failing.

```
interestpred = prediction(interestpred, loanstest$NotFullyPaid)


as.numeric(performance(interestpred, "auc")@y.values)
```

iii) Compute the test set AUC of the model. How does this compare to the first model? Which is stronger and why?

```
## [1] 0.6186048
```

- The AUC for this model is 0.619. This is worse than the model with many independent variables. We would assume that the model with all of the independent variables is stronger, but this one does fairly well with only one predictor.

C) Using the model to compute profitability

```
c = 10
r = 0.06
t = 3


c * exp(r * t)
```

i)If the loan is paid back in full, then the investor makes interest on the loan. If the loan is not paid back, the investor loses money. The investor needs to balance risk and reward. To compute interest consider a $c

investment in a loan that has an annual interest rate r over a period of t years. Using continuous compounding, the investment pays back c x e^{rt} dollars by the end of t years. How much does a $10 investment with an annual interest rate of 6% pay back after 3 years, using the interest formula?

```
## [1] 11.97217
```

ii) What is the profit to the investor if the investment is paid back in full? What if not?

- It would be c*(exp(rt)) - c. Otherwise, it would just be -c.

```
# remember 3 year term
loanstest$profit = exp(loanstest$IntRate * 3) - 1
loanstest$profit[loanstest$NotFullyPaid == 1] = -1

summary(loanstest$profit)
```

iii) Compute profit of 1 dollar investment and save to profit. It should depend on the value of NotFullyPaid. What is the max profit?

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.0000  0.2835  0.4154  0.2122  0.4984  0.8895
```

- Max profit is 8.895.

```
highinterest <- subset(loanstest, IntRate >= 0.15)
mean(highinterest$profit)
```

iv) A simple investing strategy of investing in all loans would yield a profit 20.94 for 100. This does not leverage the model we built earlier. Instead, analyze an investment strategy in which the investor only purchaes loans with a >=15% interest rate to maximize return, but with the lowest rate of failing. Model an investor who invests $1 in 100 of the best loans. Create a new dataset called HighInterest consisting of testset loans with an interest rate of at least 15%. What is the average profit? What proportion of loans were not paid back?

```
## [1] 0.2365834
```

```
riskproportion = table(highinterest$NotFullyPaid)

riskproportion[2]/sum(riskproportion)
```

```
##         1
```

```
## 0.2488688
```

- The average profit is 0.2366. Approx. 0.2489 of loans were not paid back.

```
riskpoint = sort(highinterest$PredictedRisk, decreasing = FALSE)[100]

SelectedLoans = subset(highinterest, PredictedRisk <= riskpoint)

sum(SelectedLoans$profit)
```

V) Sort the loans in HighInterest dataset by variable PredictedRisk. Create a new set called SelectedLoans that consists of the 100 loans with the smallest values of PredictedRisk. What is the profit? How many failed failed? How does this compare to the simple strategy (20.94)?

```
## [1] 31.24293
```

```
selectloans = table(SelectedLoans$NotFullyPaid)
selectloans
```

```
##
##  0  1
## 81 19
```

- Profit was 31.24. 19 of the loans failed.

```
31.24/20.94 * 100
```

```
## [1] 149.1882
```

- This is roughly 149% better than the simple strategy.

D) One of the most important assumptions of predictive modeling often does not hold in financial situations, causing predictive models to fail. What do you think this is? As an analyst, what could you do to improve the situation?

- I feel this does not account for human events, including market fluctuations. It may be better to have long-term data, especially data that is more up to date or indicative of the financial market.