DSO 530 Project

Jihyun Shin

November 22, 2015

Set-up

```
rm(list = ls())
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.1.3
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

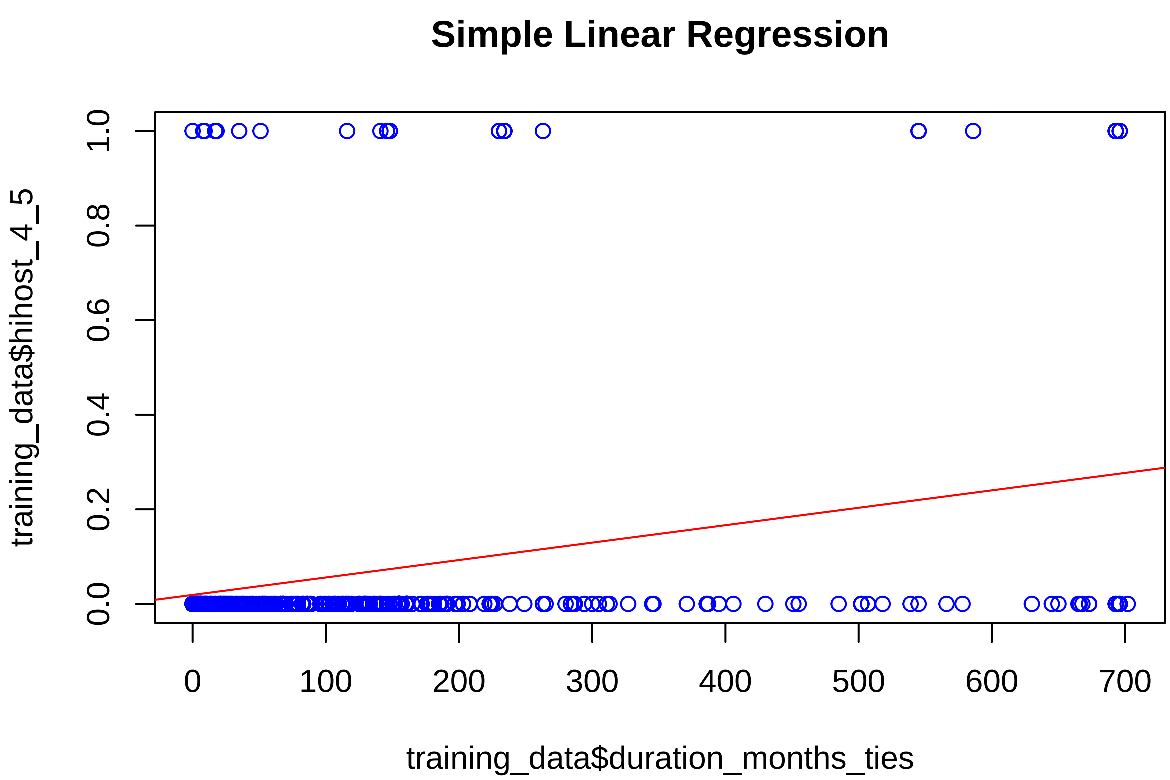
```
library(lubridate)
library(coefplot)
setwd("/Users/jihyunshin/Dropbox/USC Coursework/DSO 530 Sanctions Project")
load("merged_new_final.R")
data<-merged_new
rm(merged_new)
```

Linear Regression

```
set.seed(111)
train <- sample(1:nrow(data), nrow(data)/2)
test <- -train
training_data <- data[train,]
testing_data <- data[test,]
testing_y <- data$hihost_4_5[test]

# Simple Linear Regression

model1<-lm(training_data$hihost_4_5 ~ training_data$duration_months_ties)
summary(model1) #statistically highly significant, substantively not very strong.
```

```
##
## Call:
## lm(formula = training_data$hihost_4_5 ~ training_data$duration_months_ties)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27767 -0.07611 -0.04406 -0.02232  0.98100
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                        1.900e-02  1.492e-02   1.274    0.203
## training_data$duration_months_ties 3.685e-04  6.818e-05   5.404 1.05e-07
##
## (Intercept)
## training_data$duration_months_ties ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2477 on 456 degrees of freedom
## Multiple R-squared:  0.06019,    Adjusted R-squared:  0.05813
## F-statistic: 29.21 on 1 and 456 DF,  p-value: 1.05e-07
```

```r
plot(y=training_data$hihost_4_5,x=training_data$duration_months_ties,col="blue",main="Simple Linear Reg
abline(model1,col="red")
```




```r
# Let's add some more variables: Multiple Linear Regression
model2=lm(hihost_4_5~duration_months_ties  + ongoing_dum +
                 issue_mil_relevant_narrow+issue_mil_relevant_broad+
                  sendercosts+targetcosts+carrots_control+carrotsduringsanction_control,
```

```r
         data=training_data)
summary(model2) #Adjusted R-squared:   0.1577
```

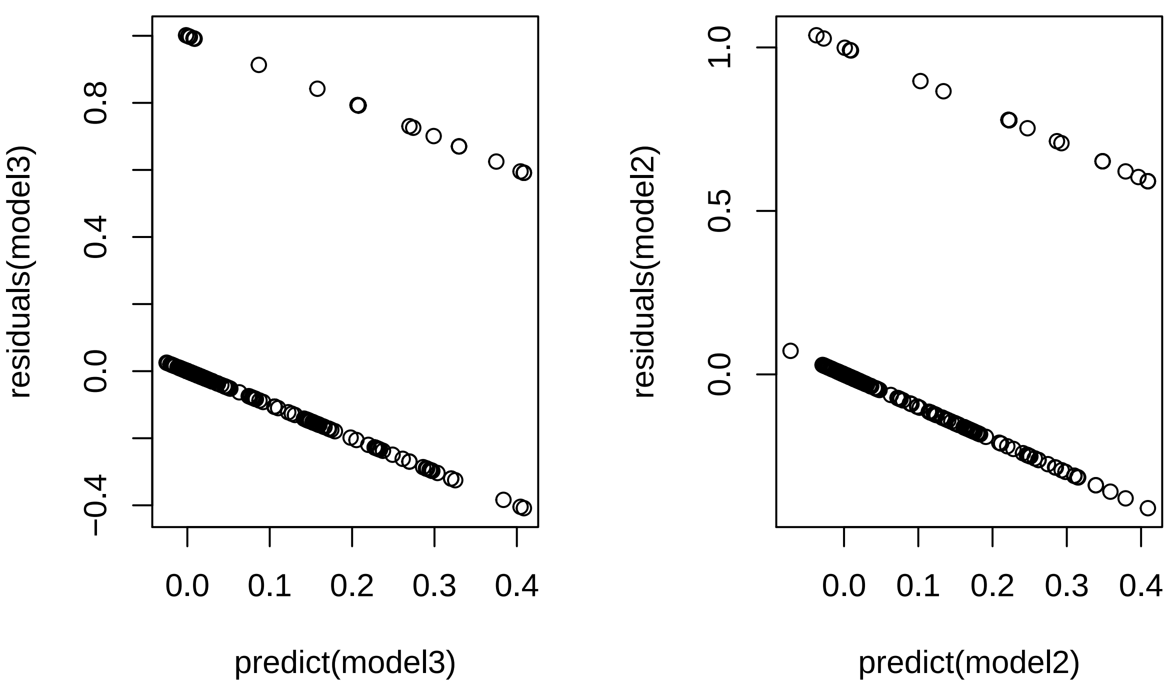
```
##
## Call:
## lm(formula = hihost_4_5 ~ duration_months_ties + ongoing_dum +
##     issue_mil_relevant_narrow + issue_mil_relevant_broad + sendercosts +
##     targetcosts + carrots_control + carrotsduringsanction_control,
##     data = training_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40919 -0.03602 -0.01169  0.00027  1.03719
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  -2.340e-01  6.343e-02  -3.689 0.000256 ***
## duration_months_ties          2.600e-04  9.486e-05   2.741 0.006398 **
## ongoing_dum1                 -2.728e-02  2.806e-02  -0.972 0.331599
## issue_mil_relevant_narrow     1.929e-02  6.400e-02   0.301 0.763261
## issue_mil_relevant_broad     -4.349e-03  6.165e-02  -0.071 0.943802
## sendercosts                   8.462e-02  5.788e-02   1.462 0.144560
## targetcosts                   1.480e-01  2.374e-02   6.236 1.15e-09 ***
## carrots_control              -4.029e-02  6.534e-02  -0.617 0.537847
## carrotsduringsanction_control -2.603e-02  3.874e-02  -0.672 0.502011
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2204 on 399 degrees of freedom
##   (50 observations deleted due to missingness)
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1577
## F-statistic: 10.52 on 8 and 399 DF,  p-value: 2.01e-13
```

```r
# The longer the sanction and the higher targetcost( but NOT the sendercost during the sanction), the m

# Let's get rid of some statistically insignificant variables
model3=lm(hihost_4_5~duration_months_ties  + ongoing_dum + sendercosts+
                 targetcosts,data=training_data)
summary(model3) #Adjusted R-squared:  0.1631
```

```
##
## Call:
## lm(formula = hihost_4_5 ~ duration_months_ties + ongoing_dum +
##     sendercosts + targetcosts, data = training_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40854 -0.03587 -0.01128  0.00023  1.00153
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -2.304e-01  6.143e-02  -3.751 0.000202 ***
```

```
## duration_months_ties  2.725e-04  9.341e-05   2.917 0.003732 **
## ongoing_dum1         -2.942e-02  2.686e-02  -1.096 0.273898
## sendercosts           8.340e-02  5.642e-02   1.478 0.140083
## targetcosts           1.455e-01  2.128e-02   6.836 3.03e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2197 on 403 degrees of freedom
##   (50 observations deleted due to missingness)
## Multiple R-squared:  0.1713, Adjusted R-squared:  0.1631
## F-statistic: 20.83 on 4 and 403 DF,  p-value: 1.27e-15
```

```r
# The result from model2 remains robust. Again, the longer the duration, and the higher the targetcosts


# Assess the linearity of the model
par(mfrow=c(1,2))
plot(predict(model3),residuals(model3))
plot(predict(model2),residuals(model2))
```




```r
par(mfrow=c(1,1))
# There is a very strong and specific pattern: linear regression is a bad model for our data.


# Introduce Interactions
model4=lm(hihost_4_5~duration_months_ties*targetcosts +
                  ongoing_dum,data=training_data)
summary(model4)
```

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

```
## lm(formula = hihost_4_5 ~ duration_months_ties * targetcosts +
##     ongoing_dum, data = training_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66849 -0.02756 -0.02351 -0.00999  0.98957
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -0.0645335  0.0369328  -1.747 0.081324
## duration_months_ties             -0.0005012  0.0001943  -2.579 0.010247
## targetcosts                       0.0922401  0.0252120   3.659 0.000286
## ongoing_dum1                     -0.0101457  0.0263070  -0.386 0.699944
## duration_months_ties:targetcosts  0.0004523  0.0001107   4.085 5.29e-05
##
## (Intercept)                      .
## duration_months_ties             *
## targetcosts                      ***
## ongoing_dum1
## duration_months_ties:targetcosts ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2187 on 413 degrees of freedom
##   (40 observations deleted due to missingness)
## Multiple R-squared:  0.1901, Adjusted R-squared:  0.1822
## F-statistic: 24.23 on 4 and 413 DF,  p-value: < 2.2e-16
```

```
# The effect of target cost proves to be robust again. The higher the target cost, the more likely we s
```

Discussion:From the linear regression, we can infer that the target cost is a very significant factor. However, as we have seen in the pattern between residuals and predicted values, linear regression is a bad model for our data. This is partly because our DV is a binary variable, for which logistic regression may be a better model. Let's now turn to logistic regression.

Simple Logistic Regression

DV: hihost__4__5 (War & Use of force coded as 1; No militarized action &Threat to use force & Display of force coded as 0)

```
set.seed(1)
train <- sample(1:nrow(data), nrow(data)/2)
test <- -train
training_data <- data[train,]
testing_data <- data[test,]
testing_y <- data$hihost_4_5[test]

mod_logit <- glm(hihost_4_5 ~ duration_months_ties  + ongoing_dum +
                 issue_mil_relevant_narrow+issue_mil_relevant_broad+sendercosts+targetcosts+carrots_con
```

```r
                 data = training_data,
                 family = "binomial")
summary(mod_logit)
```

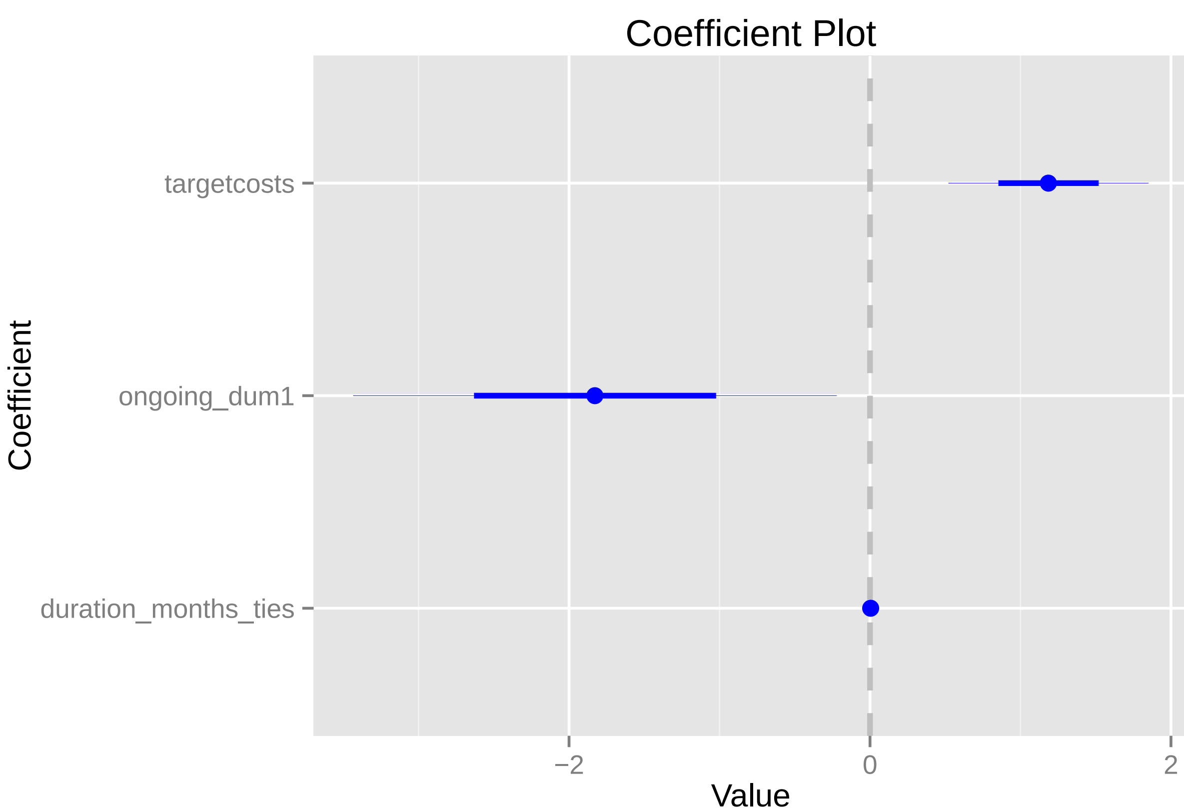
```
##
## Call:
## glm(formula = hihost_4_5 ~ duration_months_ties + ongoing_dum +
##     issue_mil_relevant_narrow + issue_mil_relevant_broad + sendercosts +
##     targetcosts + carrots_control + carrotsduringsanction_control,
##     family = "binomial", data = training_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4227  -0.2699  -0.2000  -0.1066   2.8490
##
## Coefficients:
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                   -5.967876   1.021448  -5.843 5.14e-09 ***
## duration_months_ties           0.004437   0.001698   2.613  0.00899 **
## ongoing_dum1                  -1.880268   0.871233  -2.158  0.03091 *
## issue_mil_relevant_narrow     -0.188536   0.932627  -0.202  0.83979
## issue_mil_relevant_broad       0.500625   1.031684   0.485  0.62750
## sendercosts                    0.974919   0.727118   1.341  0.17999
## targetcosts                    1.051093   0.375145   2.802  0.00508 **
## carrots_control                0.989531   0.944975   1.047  0.29503
## carrotsduringsanction_control -0.130270   0.738948  -0.176  0.86007
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 159.42  on 405  degrees of freedom
## Residual deviance: 124.34  on 397  degrees of freedom
##   (52 observations deleted due to missingness)
## AIC: 142.34
##
## Number of Fisher Scoring iterations: 7
```

```r
#Let's get rid of the statistically insignificant predictors.
mod_logit2 <- glm(hihost_4_5 ~ duration_months_ties  + ongoing_dum+ sendercosts+targetcosts,
                 data = training_data,
                 family = "binomial")
summary(mod_logit2)
```

```
##
## Call:
## glm(formula = hihost_4_5 ~ duration_months_ties + ongoing_dum +
##     sendercosts + targetcosts, family = "binomial", data = training_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3097  -0.2595  -0.2102  -0.1152   2.7700
##
```

```
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -6.128600   0.999043  -6.134 8.54e-10 ***
## duration_months_ties  0.004420   0.001579   2.800 0.005106 **
## ongoing_dum1         -1.828136   0.802432  -2.278 0.022712 *
## sendercosts           1.123962   0.690449   1.628 0.103552
## targetcosts           1.185435   0.331666   3.574 0.000351 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 159.42  on 405  degrees of freedom
## Residual deviance: 125.62  on 401  degrees of freedom
##   (52 observations deleted due to missingness)
## AIC: 135.62
##
## Number of Fisher Scoring iterations: 7
```

```
# coefficient plots without the intercept;
coefplot(mod_logit2, coefficients=c("ongoing_dum1","duration_months_ties","targetcosts"))
```



Discussion: The result shows that the longer the sanction, the higher the likelihood (0.0044) of war / use of force (p<0.01). However, if the sanction is ongoing, note that there is a substantially lower likelihood (-1.8) that the sanction involved any use of force and war (p<0.05). Again, the effect of target costs remains robust.

```
# Let's use the testing_data and calculate the error rate.
mod_logit2_probs = predict(mod_logit2, testing_data, type = "response")
head(mod_logit2_probs)
```

```
##          1          2          3          4          5          7
## 0.06952646 0.07893125         NA 0.04464522 0.02166057 0.07281027
```

```
logistic_pred_y = rep("0", length(testing_y))
logistic_pred_y[mod_logit2_probs > 0.5] = "1"

conf_matrix = table(testing_y, logistic_pred_y)
conf_matrix
```

```
##          logistic_pred_y
## testing_y   0   1
##         0 417   2
##         1  38   2
```

```
error_rate = 40/ (417+2+2+38)
error_rate
```

```
## [1] 0.08714597
```

```
# OR
logit<-mean(testing_y != logistic_pred_y)
logit
```

```
## [1] 0.08714597
```

Logistic regression with a threshold of 0.5 yields an error rate of 0.08714597

LDA

```
library(MASS) # Use MASS library for LDA function
```

```
## Warning: package 'MASS' was built under R version 3.1.3

##
## Attaching package: 'MASS'
##
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
lda_model=lda(hihost_4_5 ~ duration_months_ties  + ongoing_dum,
                  data=training_data)
names(lda_model)
```

```
##  [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"     "N"
##  [8] "call"    "terms"   "xlevels"
```

```r
lda_predict=predict(lda_model, testing_data)
names(lda_predict)
```

```
## [1] "class"     "posterior" "x"
```

```r
lda_predicted_y=lda_predict$class
head(lda_predicted_y)
```

```
## [1] 0 0 1 0 0 1
## Levels: 0 1
```

```r
# confusion matrix
table(testing_y, lda_predicted_y)
```

```
##          lda_predicted_y
## testing_y   0   1
##         0 407  12
##         1  27  13
```

```r
lda<-(12+27)/(407+12+27+13) #  0.08496732
lda
```

```
## [1] 0.08496732
```

LDA yields an error rate of 0.08496732.

QDA

```r
qda_model=qda(hihost_4_5 ~ duration_months_ties  + ongoing_dum, data=training_data)
names(qda_model)
```

```
##  [1] "prior"   "counts"  "means"   "scaling" "ldet"    "lev"     "N"
##  [8] "call"    "terms"   "xlevels"
```

```r
qda_predict=predict(qda_model, testing_data)
names(qda_predict)
```

```
## [1] "class"     "posterior"
```

```r
qda_predicted_y=qda_predict$class
head(qda_predicted_y)
```

```
## [1] 0 0 0 0 0 1
## Levels: 0 1
```

```r
# confusion matrix
table(testing_y, qda_predicted_y)
```

```
##          qda_predicted_y
## testing_y   0   1
##         0 409  10
##         1  21  19
```

```r
qda<-(10+21)/(409+10+21+19)
qda
```

```
## [1] 0.06753813
```

QDA yields an error rate of 0.06753813.

Comparison between Logit, LDA, QDA

```r
library(pander)
```

```
## Warning: package 'pander' was built under R version 3.1.3
```

```r
Models<-c("Logistic Regression","LDA","QDA")
Error_Rate<-c(logit, lda, qda)
tab_comp<-rbind(Models, Error_Rate)
pander(tab_comp)
```

Table 1: Table continues below

Models	Logistic Regression	LDA
Error_Rate	0.0871459694989107	0.0849673202614379

Models	QDA
Error_Rate	0.0675381263616558

QDA is the winner!