

# Homework 3: The Application of Logistic Regression to Examine the Predictors of Car Crashes Caused by Alcohol

Yiming Cao, Sujan Kakumanu, Angel Sanaa Rutherford

2025-11-20

```
mydata <- read.csv("Logistic Regression Data.csv")

DRINKING_D.tab <- table(mydata$DRINKING_D)
prop.table(DRINKING_D.tab) #94% of crashes did not involve drunk driver while 5.75 did
```

	0	1
	0.9426944	0.0573056

```
CrossTable(mydata$DRINKING_D, mydata$FATAL_OR_M, prop.r=FALSE,prop.chisq=FALSE, chisq=FALSE,)
```

Cell Contents	
-----	
	N
N / Col Total	
-----	

Total Observations in Table: 43364

	mydata\$FATAL_OR_M		
mydata\$DRINKING_D	0	1	Row Total

0	39698	1181	40879
	0.945	0.863	
1	2297	188	2485
	0.055	0.137	
Column Total	41995	1369	43364
	0.968	0.032	

```
CrossTable(mydata$DRINKING_D, mydata$OVERTURNED, prop.r=FALSE,prop.chisq=FALSE, chisq=FALSE,)
```

Cell Contents	
	N
N / Col Total	

Total Observations in Table: 43364

mydata\$DRINKING_D	mydata\$OVERTURNED		Row Total
	0	1	
0	40267	612	40879
	0.944	0.848	
1	2375	110	2485
	0.056	0.152	
Column Total	42642	722	43364
	0.983	0.017	

```
CrossTable(mydata$DRINKING_D, mydata$CELL_PHONE ,prop.r=FALSE,prop.chisq=FALSE, chisq=FALSE,pr
```

```

      Cell Contents
|-----|
|              N |
|      N / Col Total |
|-----|

```

Total Observations in Table: 43364

	mydata\$CELL_PHONE		
mydata\$DRINKING_D	0	1	Row Total
0	40453	426	40879
	0.943	0.938	
1	2457	28	2485
	0.057	0.062	
Column Total	42910	454	43364
	0.990	0.010	

```
CrossTable(mydata$DRINKING_D, mydata$SPEEDING ,prop.r=FALSE,prop.chisq=FALSE, chisq=FALSE,pr
```

```

      Cell Contents
|-----|
|              N |
|      N / Col Total |
|-----|

```

Total Observations in Table: 43364

	mydata\$SPEEDING		
mydata\$DRINKING_D	0	1	Row Total
0	39618	1261	40879
	0.947	0.829	
1	2225	260	2485
	0.053	0.171	
Column Total	41843	1521	43364
	0.965	0.035	

```
CrossTable(mydata$DRINKING_D, mydata$AGGRESSIVE ,prop.r=FALSE,prop.chisq=FALSE, chisq=FALSE,
```

#### Cell Contents

	N
N / Col Total	

Total Observations in Table: 43364

	mydata\$AGGRESSIVE		
mydata\$DRINKING_D	0	1	Row Total
0	22357	18522	40879
	0.934	0.953	
1	1569	916	2485
	0.066	0.047	
Column Total	23926	19438	43364
	0.552	0.448	

-----|-----|-----|-----|

```
CrossTable(mydata$DRINKING_D, mydata$DRIVER1617 ,prop.r=FALSE,prop.chisq=FALSE, chisq=FALSE,
```

Cell Contents

-----	
	N
N / Col Total	
-----	

Total Observations in Table: 43364

	mydata\$DRIVER1617		
mydata\$DRINKING_D	0	1	Row Total
----- ----- ----- -----			
0	40205	674	40879
	0.942	0.983	
----- ----- ----- -----			
1	2473	12	2485
	0.058	0.017	
----- ----- ----- -----			
Column Total	42678	686	43364
	0.984	0.016	
----- ----- ----- -----			

```
CrossTable(mydata$DRINKING_D, mydata$DRIVER65PLUS ,prop.r=FALSE,prop.chisq=FALSE, chisq=FALSE,
```

Cell Contents

-----	
	N
N / Col Total	

|-----|

Total Observations in Table: 43364

		mydata\$DRIVER65PLUS		
mydata\$DRINKING_D		0	1	Row Total
	0	36642	4237	40879
		0.939	0.973	
	1	2366	119	2485
		0.061	0.027	
Column Total		39008	4356	43364
		0.900	0.100	

```
CrossTable(mydata$DRINKING_D, mydata$FATAL_OR_M, prop.r=FALSE,prop.chisq=FALSE, chisq=TRUE,p
```

Cell Contents

	N	
	N / Col Total	

Total Observations in Table: 43364

		mydata\$FATAL_OR_M		
mydata\$DRINKING_D		0	1	Row Total
	0	39698	1181	40879
		0.945	0.863	
	1	2297	188	2485



1	2375	110	2485
	1.927	113.824	
	0.056	0.152	
----- ----- ----- -----			
Column Total	42642	722	43364
	0.983	0.017	
----- ----- ----- -----			

```
CrossTable(mydata$DRINKING_D, mydata$CELL_PHONE ,prop.r=FALSE,prop.chisq=FALSE, chisq=TRUE,p
```

Cell Contents
-----
N
N / Col Total
-----

Total Observations in Table: 43364

	mydata\$CELL_PHONE		
mydata\$DRINKING_D	0	1	Row Total
----- ----- ----- -----			
0	40453	426	40879
	0.943	0.938	
----- ----- ----- -----			
1	2457	28	2485
	0.057	0.062	
----- ----- ----- -----			
Column Total	42910	454	43364
	0.990	0.010	
----- ----- ----- -----			

Statistics for All Table Factors

Pearson's Chi-squared test



-----  
Chi^2 = 0.162071      d.f. = 1      p = 0.6872569

Pearson's Chi-squared test with Yates' continuity correction  
-----

Chi^2 = 0.09065262      d.f. = 1      p = 0.7633491

```
CrossTable(mydata$DRINKING_D, mydata$SPEEDING ,prop.r=FALSE,prop.chisq=FALSE, chisq=TRUE,prop
```

#### Cell Contents

```
|-----|  
|                      N |  
|          N / Col Total |  
|-----|
```

Total Observations in Table: 43364

	mydata\$SPEEDING		
mydata\$DRINKING_D	0	1	Row Total
0	39618	1261	40879
	0.947	0.829	
1	2225	260	2485
	0.053	0.171	
Column Total	41843	1521	43364
	0.965	0.035	

Statistics for All Table Factors

Pearson's Chi-squared test  
-----



Pearson's Chi-squared test with Yates' continuity correction

-----  
Chi^2 = 67.2607      d.f. = 1      p = 0.0000000000000002378758

```
CrossTable(mydata$DRINKING_D, mydata$DRIVER1617 ,prop.r=FALSE,prop.chisq=FALSE, chisq=TRUE,p
```

```
Cell Contents
|-----|
|              N |
|      N / Col Total |
|-----|
```

Total Observations in Table: 43364

	mydata\$DRIVER1617		
mydata\$DRINKING_D	0	1	Row Total
0	40205	674	40879
	0.942	0.983	
1	2473	12	2485
	0.058	0.017	
Column Total	42678	686	43364
	0.984	0.016	

Statistics for All Table Factors

Pearson's Chi-squared test

-----  
Chi^2 = 20.45167      d.f. = 1      p = 0.000006115619

-----

Chi^2 = 19.7097      d.f. = 1      p = 0.000009014275

```
CrossTable(mydata$DRINKING_D, mydata$DRIVER65PLUS ,prop.r=FALSE,prop.chisq=FALSE, chisq=TRUE
```

Cell Contents	
	N
N / Col Total	

Total Observations in Table: 43364

		mydata\$DRIVER65PLUS		
mydata\$DRINKING_D		0	1	Row Total
	0	36642	4237	40879
		0.939	0.973	
	1	2366	119	2485
		0.061	0.027	
	Column Total	39008	4356	43364
		0.900	0.100	

### Statistics for All Table Factors

Pearson's Chi-squared test

---

Chi^2 = 80.6047      d.f. = 1      p = 0.000000000000000000275703

Pearson's Chi-squared test with Yates' continuity correction

-----  
Chi^2 = 79.9888      d.f. = 1      p = 0.000000000000000003765375

```
tapply(mydata$PCTBACHMOR,  
mydata$DRINKING_D, mean)
```

0	1
16.56986	16.61173

```
tapply(mydata$PCTBACHMOR, mydata$DRINKING_D, sd)
```

0	1
18.21426	18.72091

```
tapply(mydata$MEDHHINC,  
mydata$DRINKING_D, mean)
```

0	1
31483.05	31998.75

```
tapply(mydata$MEDHHINC, mydata$DRINKING_D, sd)
```

0	1
16930.1	17810.5

```
t.test(mydata$PCTBACHMOR~mydata$DRINKING_D)
```

Welch Two Sample t-test

data: mydata\$PCTBACHMOR by mydata\$DRINKING\_D

t = -0.10842, df = 2777.5, p-value = 0.9137

alternative hypothesis: true difference in means between group 0 and group 1 is not equal to  
95 percent confidence interval:

-0.7991398 0.7153982

sample estimates:

mean in group 0	mean in group 1
16.56986	16.61173

```
t.test(mydata$MEDHHINC~mydata$DRINKING_D)
```

Welch Two Sample t-test

```
data: mydata$MEDHHINC by mydata$DRINKING_D
t = -1.4053, df = 2763.9, p-value = 0.16
alternative hypothesis: true difference in means between group 0 and group 1 is not equal to
95 percent confidence interval:
 -1235.2508  203.8544
sample estimates:
mean in group 0 mean in group 1
    31483.05      31998.75
```

```
cor(mydata$PCTBACHMOR, mydata$DRINKING_D, method="pearson")
```

```
[1] 0.0005334492
```

```
cor(mydata$MEDHHINC, mydata$DRINKING_D, method="pearson")
```

```
[1] 0.007058232
```

```
cor(mydata$FATAL_OR_M, mydata$DRINKING_D, method="pearson")
```

```
[1] 0.06216164
```

```
cor(mydata$OVERTURNED, mydata$DRINKING_D, method="pearson")
```

```
[1] 0.05321245
```

```
cor(mydata$CELL_PHONE, mydata$DRINKING_D, method="pearson")
```

```
[1] 0.00193325
```

```
cor(mydata$SPEEDING, mydata$DRINKING_D, method="pearson")
```

```
[1] 0.09321369
```

```
cor(mydata$AGGRESSIVE, mydata$DRINKING_D, method="pearson")
```

```
[1] -0.03948341
```

```
cor(mydata$DRIVER1617, mydata$DRINKING_D, method="pearson")
```

```
[1] -0.02171699
```

```
cor(mydata$DRIVER65PLUS, mydata$DRINKING_D, method="pearson")
```

```
[1] -0.04311372
```

```
#no severe multicollinearity :)
```

```
full_logit <- glm(DRINKING_D ~ FATAL_OR_M +  
OVERTURNED + CELL_PHONE + SPEEDING + AGGRESSIVE + DRIVER1617 +  
DRIVER65PLUS + PCTBACHMOR + MEDHHINC, data = mydata, family = "binomial")
```

```
full_logit_output <- summary(full_logit)  
full_logit_output
```

Call:

```
glm(formula = DRINKING_D ~ FATAL_OR_M + OVERTURNED + CELL_PHONE +  
SPEEDING + AGGRESSIVE + DRIVER1617 + DRIVER65PLUS + PCTBACHMOR +  
MEDHHINC, family = "binomial", data = mydata)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-2.732506616	0.045875659	-59.563	< 0.0000000000000002 ***
FATAL_OR_M	0.814013802	0.083806924	9.713	< 0.0000000000000002 ***
OVERTURNED	0.928921376	0.109166324	8.509	< 0.0000000000000002 ***
CELL_PHONE	0.029550085	0.197777821	0.149	0.8812
SPEEDING	1.538975665	0.080545894	19.107	< 0.0000000000000002 ***
AGGRESSIVE	-0.596915946	0.047779238	-12.493	< 0.0000000000000002 ***
DRIVER1617	-1.280295964	0.293147168	-4.367	0.000012572447127933 ***
DRIVER65PLUS	-0.774664640	0.095858315	-8.081	0.000000000000000641 ***
PCTBACHMOR	-0.000370634	0.001296387	-0.286	0.7750





[illegible]

Cell Contents	
	N
N / Col Total	

	mydata\$DRINKING_D		
fit.binary	0	1	Row Total
FALSE	2374	41	2415
	0.058	0.016	
TRUE	38505	2444	40949
	0.942	0.984	
Column Total	40879	2485	43364
	0.943	0.057	

```
-----|-----|-----|-----|
```

```
fit.binary = (fit>=0.03)
CrossTable(fit.binary, mydata$DRINKING_D, prop.r=FALSE, prop.t=FALSE, prop.chisq=FALSE)
```

```

Cell Contents
|-----|
|              N |
|      N / Col Total |
|-----|

```

Total Observations in Table: 43364

	mydata\$DRINKING_D		
fit.binary	0	1	Row Total
FALSE	2613	48	2661
	0.064	0.019	
TRUE	38266	2437	40703
	0.936	0.981	
Column Total	40879	2485	43364
	0.943	0.057	

```
fit.binary = (fit>=0.05)
CrossTable(fit.binary, mydata$DRINKING_D, prop.r=FALSE, prop.t=FALSE, prop.chisq=FALSE)
```

```

Cell Contents
|-----|

```

	N
N / Col Total	
-----	

Total Observations in Table: 43364

	mydata\$DRINKING_D		
fit.binary	0	1	Row Total
-----	-----	-----	-----
FALSE	19176	659	19835
	0.469	0.265	
-----	-----	-----	-----
TRUE	21703	1826	23529
	0.531	0.735	
-----	-----	-----	-----
Column Total	40879	2485	43364
	0.943	0.057	
-----	-----	-----	-----

```
fit.binary = (fit>=0.07)
CrossTable(fit.binary, mydata$DRINKING_D, prop.r=FALSE, prop.t=FALSE, prop.chisq=FALSE)
```

Cell Contents
-----
N
N / Col Total
-----

Total Observations in Table: 43364

	mydata\$DRINKING_D		
fit.binary	0	1	Row Total
-----	-----	-----	-----
FALSE	37356	1935	39291

	0.914	0.779	
TRUE	3523	550	4073
	0.086	0.221	
Column Total	40879	2485	43364
	0.943	0.057	

```
fit.binary = (fit>=0.08)
CrossTable(fit.binary, mydata$DRINKING_D, prop.r=FALSE, prop.t=FALSE, prop.chisq=FALSE)
```

Cell Contents

N
N / Col Total

Total Observations in Table: 43364

	mydata\$DRINKING_D		
fit.binary	0	1	Row Total
FALSE	38370	2026	40396
	0.939	0.815	
TRUE	2509	459	2968
	0.061	0.185	
Column Total	40879	2485	43364
	0.943	0.057	

```
fit.binary = (fit>=0.09)
CrossTable(fit.binary, mydata$DRINKING_D, prop.r=FALSE, prop.t=FALSE, prop.chisq=FALSE)
```

```

      Cell Contents
|-----|
|              N |
|      N / Col Total |
|-----|

```

Total Observations in Table: 43364

	mydata\$DRINKING_D		
fit.binary	0	1	Row Total
FALSE	38670	2067	40737
	0.946	0.832	
TRUE	2209	418	2627
	0.054	0.168	
Column Total	40879	2485	43364
	0.943	0.057	

```
fit.binary = (fit>=0.1)
CrossTable(fit.binary, mydata$DRINKING_D, prop.r=FALSE, prop.t=FALSE, prop.chisq=FALSE)
```

```

      Cell Contents
|-----|
|              N |
|      N / Col Total |
|-----|

```

Total Observations in Table: 43364

	mydata\$DRINKING_D		
fit.binary	0	1	Row Total
FALSE	38762	2077	40839
	0.948	0.836	
TRUE	2117	408	2525
	0.052	0.164	
Column Total	40879	2485	43364
	0.943	0.057	

```
fit.binary = (fit>=0.15)
CrossTable(fit.binary, mydata$DRINKING_D, prop.r=FALSE, prop.t=FALSE, prop.chisq=FALSE)
```

Cell Contents

	N
N / Col Total	

Total Observations in Table: 43364

	mydata\$DRINKING_D		
fit.binary	0	1	Row Total
FALSE	39743	2226	41969
	0.972	0.896	
TRUE	1136	259	1395
	0.028	0.104	

----- ----- ----- -----				
Column Total	40879	2485	43364	
	0.943	0.057		
----- ----- ----- -----				

```
fit.binary = (fit>=0.2)
CrossTable(fit.binary, mydata$DRINKING_D, prop.r=FALSE, prop.t=FALSE, prop.chisq=FALSE)
```

Cell Contents

-----	
	N
	N / Col Total
-----	

Total Observations in Table: 43364

	mydata\$DRINKING_D		
fit.binary	0	1	Row Total
-----	-----	-----	-----
FALSE	40690	2428	43118
	0.995	0.977	
-----	-----	-----	-----
TRUE	189	57	246
	0.005	0.023	
-----	-----	-----	-----
Column Total	40879	2485	43364
	0.943	0.057	
-----	-----	-----	-----

```
fit.binary = (fit>=0.5)
CrossTable(fit.binary, mydata$DRINKING_D, prop.r=FALSE, prop.t=FALSE, prop.chisq=FALSE)
```

Cell Contents	
	N
	N / Col Total

Total Observations in Table: 43364

fit.binary	mydata\$DRINKING_D		Row Total
	0	1	
FALSE	40875	2481	43356
	1.000	0.998	
TRUE	4	4	8
	0.000	0.002	
Column Total	40879	2485	43364
	0.943	0.057	

```
a <- cbind(mydata$DRINKING_D, fit)

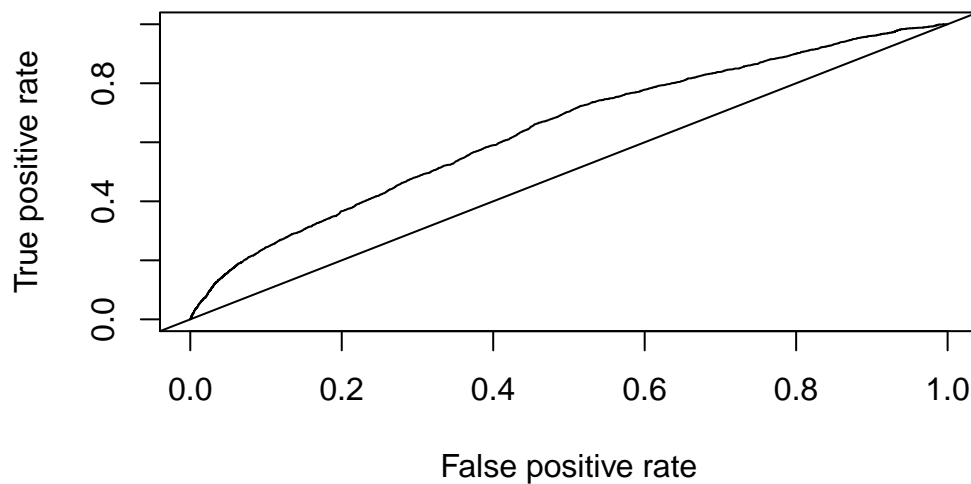
colnames(a) <- c("labels", "predictions")

roc <- as.data.frame(a)

pred <- prediction(roc$predictions, roc$labels)

roc.perf = performance(pred, measure = "tpr", x.measure="fpr")
plot(roc.perf)
abline(a=0,b=1)
```





```
opt.cut = function(perf, pred){
  cut.ind = mapply(FUN=function(x, y, p){
    d = (x - 0)^2 + (y-1)^2
    ind = which(d == min(d))
    c(sensitivity = y[[ind]], specificity = 1-x[[ind]],
      cutoff = p[[ind]])
  }, perf@x.values, perf@y.values, pred@cutoffs)
}
```

```
print(opt.cut(roc.perf, pred))
```

```
      [,1]
sensitivity 0.66076459
specificity 0.54524328
cutoff      0.06365151
```

```
auc.perf = performance(pred, measure ="auc")
auc.perf@y.values #statisticians says that area >.7 is acceptable
```

```
[[1]]
[1] 0.6398695
```

```
binary_logit <- glm(DRINKING_D ~ FATAL_OR_M +
OVERTURNED + CELL_PHONE + SPEEDING + AGGRESSIVE + DRIVER1617 +
DRIVER65PLUS, data = mydata, family = "binomial")
```

```
binary_logit_output <- summary(binary_logit)
binary_logit_output
```

Call:

```
glm(formula = DRINKING_D ~ FATAL_OR_M + OVERTURNED + CELL_PHONE +
    SPEEDING + AGGRESSIVE + DRIVER1617 + DRIVER65PLUS, family = "binomial",
    data = mydata)
```

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-2.65190	0.02753	-96.324	< 0.0000000000000002 ***
FATAL_OR_M	0.80932	0.08376	9.662	< 0.0000000000000002 ***
OVERTURNED	0.93978	0.10903	8.619	< 0.0000000000000002 ***
CELL_PHONE	0.03107	0.19777	0.157	0.875
SPEEDING	1.54032	0.08053	19.128	< 0.0000000000000002 ***
AGGRESSIVE	-0.59365	0.04775	-12.433	< 0.0000000000000002 ***
DRIVER1617	-1.27158	0.29311	-4.338	0.00001436374143265 ***
DRIVER65PLUS	-0.76646	0.09576	-8.004	0.00000000000000121 ***

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 19036 on 43363 degrees of freedom  
Residual deviance: 18344 on 43356 degrees of freedom  
AIC: 18360

Number of Fisher Scoring iterations: 6

```
or_ci <- exp(cbind(OR = coef(binary_logit), confint(binary_logit)))
```

Waiting for profiling to be done...



# 1 Introduction

## 2 Methods

### 2.1 a) + b) - Sujan

### 2.2 c) + d) - Angel

In logistic regression, each predictor  $x_i$  is tested for the null hypothesis,  $H_0$ , that the beta coefficient,  $\beta_i$ , is 0 against the alternative hypothesis  $H_a$  that  $\beta_i$  is not 0:

$$H_0 : \beta_i = 0$$

$$H_a : \beta_i \neq 0$$

The z-value, also known as the Wald statistic in logistic regression, is the test statistic that we calculate under the null hypothesis. We calculate this statistic by dividing the estimated beta coefficient,  $\hat{\beta}_i$ , by its standard error or  $\sigma_{\hat{\beta}_i}$ :

$$z = \frac{\hat{\beta}_i}{\sigma_{\hat{\beta}_i}}$$

Under the null hypothesis, the Wald statistic follows an approximately standard normal distribution,  $N(0,1)$ . This property allows us to compute the two-tailed p-value as the probability of observing a statistic as extreme, or more extreme, than the calculated statistic if the null hypothesis were true. If the p-value is  $< 0.05$ , we can reject the null hypothesis in favor of the alternative hypothesis that  $\beta_i$  is not 0. Rather than interpreting the raw beta coefficients, statisticians prefer use the odds ratio,  $OR_i$ , which can be calculated by exponentiating  $\hat{\beta}_i$ :

$$OR_i = e^{\hat{\beta}_i}$$

The odds ratio expresses the effect of a predictor on the dependent variable in multiplicative terms. Specifically, it represents how the odds of the event change for a one-unit increase in the predictor, holding other variables constant. The null and alternative hypothesis can be adapted for the odds ratio, where the null hypothesis is the predictor has no effect on the odds ( $OR = 1$ ) and the alternative hypothesis is that the predictor increases or decrease the odds of the event ( $OR \neq 1$ ):

$$H_0 : OR = 1$$

$$H_a : OR \neq 1$$

Conceptually, the odds ratio is the ratio of the odds with the predictor present to the odds with the predictor absent. Thus, if the odds ratio equals 1, it indicates that the odds are

the same: the predictor did not change the odds of the outcome. Alternatively, if the odds ratio is significantly above or below 1, the predictor increased or decreased the odds. The confidence intervals for the odds ratios can be calculated by exponentiating the coefficient confidence intervals. These intervals provide a range of plausible values for the true odds ratio, reflecting the uncertainty of the estimate. In the context of logistic regression, the presence of a 1 in the confidence interval indicates the predictor's effect is not statistically significant while a confidence interval entirely above or below 1, indicates that the predictor increased or decreased the odds.

All coefficient estimates, z-values, and p-values were extracted in R from the fitted logistic regression model's summary. Odds ratios and their confidence intervals were calculated by exponentiating the original coefficient estimates and confidence intervals, then merged with the extracted coefficients for interpretation.

In our analysis, goodness of the model's fit was evaluated in various ways. In Ordinary Least Squares (OLS) regression,  $R^2$  is used to evaluate model fit as it is a statistic that returns the proportion of total variance in the dependent variable explained by the independent variable. Unlike in OLS regression, logistic regression doesn't model a continuous outcome. In logistic regression the dependent variable,  $Y$  is binary, taking a value of 1 to indicate the occurrence of an event or 0 to indicate its absence. Therefore, since there is no longer a meaningful attribution of unexplained and explained variance in the dependent variable,  $R^2$  can no longer be interpreted as the percent of variance explained by the model. Similarly to linear regression, residuals,  $\varepsilon_i$ , are calculated as the difference between the observed values of the dependent variable,  $y_i$ , and the predicted values of the dependent variable,  $\hat{y}_i$ :

$$\varepsilon_i = y_i - \hat{y}_i$$

In logistic regression, however, the predicted values,  $\hat{y}_i$ , represent the probability that  $Y = 1$ , while  $y_i$  represent the binary outcome ( $Y = 1$  or  $Y = 0$ ). Thus, residuals represent the difference between the observed binary outcome and the model's predicted probabilities. Theoretically a model of good fit predicts high probabilities of  $Y = 1$  if  $y_i$  actually equals 1 and a low probability of  $Y = 1$  if  $y_i$  is actually 0. In order to determine what is considered high probability and low probability, a cut-off value is imposed on the  $\hat{y}_i$  values. Cut-off values are then evaluated based on their specificity, sensitivity, and misclassification rates. Sensitivity, also called the true positive rate, is the proportion of actual positives that are correctly identified:

$$\text{Sensitivity} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

In this analysis, the sensitivity rate is the proportion of observed  $y_i = 1$  values correctly predicted as 1. Specificity, also called the true negative rate, is the proportion of actual negatives that are correctly identified as negatives:

$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}}$$

The specificity rate in this analysis is the proportion of observed  $y_i = 0$  values correctly predicted as 0. The misclassification rate is the proportion of incorrectly identified positive and negative  $y_i$  values based on the total number of predictions:

$$\text{Misclassification} = \frac{\text{False Negatives} + \text{False Positives}}{\text{True Negatives} + \text{True Positives} + \text{False Positives} + \text{False Negatives}}$$

In R, we called upon `fit.binary` and set the fit parameter to various different values to simulate how various cut off values would impact the sensitivity, specification, and misclassification rate. In other words, we use multiple cut-off values to compare the trade-offs of each cut-off threshold. Ideally, the chosen threshold will achieve higher sensitivity and specificity while minimizing the misclassification rate.

Receiver Operating Characteristics (ROC) curves are another tool for evaluating cut-off values. The ROC curve plots sensitivity against the false positive rate ( $1 - \text{specificity}$ ) across all possible cut-off values of  $\hat{y}_i$ . The baseline for evaluating ROC curves called the “worthless” ROC is a 45 degree line where sensitivity and the false positive rate are equal across all cut-off values, meaning the predictions are no better than a random guess. Effective models produce ROC curves that lie above this diagonal baseline. ROC curves can be used to determine the cut-off value that balances the sensitivity and specificity rate, characteristics that indicate a good model. One common way to determine the optimal cut-off value is to use the Youden Index, which identifies the cut-off that maximizes the sum of sensitivity and specificity is maximized:

$$J = \text{Sensitivity} + \text{Specificity} - 1$$

This corresponds to the point on the ROC curve farthest above the diagonal line, or equivalently, the point closest to the top-left corner of the graph where sensitivity and specificity both equal 1. To identify the optimal cut-off value, we implemented a function in R that is conceptually similar to the Youden Index as it attempts to find the point that minimizes the distance to this ideal point.

In addition to identifying an optimal cut-off, we can also calculate Area Under Curve (AUC) for our ROC curve as a measure of the model’s overall predictive accuracy. The AUC quantifies the model’s ability to discriminate between positive and negative outcomes across all possible cut-offs. An AUC of 1 (area of the entire graph) indicates perfect classification or discrimination while a value of 0.5 (area under the 45 degree line) indicates no better than random guessing. Higher AUC values indicate that the model achieves strong predictive accuracy regardless of any single cut-off value, which in turn means that there exists at least one cut-off value where both sensitivity and specificity are relatively high. In this analysis, the AUC was computed in R using the `performance` function from the `ROCR` package. We relied on commonly established thresholds for evaluating model accuracy based on AUC values where 0.90–1.00 indicates excellent accuracy, 0.80–0.90 good, 0.70–0.80 fair, 0.60–0.70 poor, and 0.50–0.60 indicates the model failed.

Another measure used to evaluate logistic regression model fit is the Akaike Information Criterion (AIC). Although the absolute value of the AIC is not interpretable on its own, it provides

a basis for comparing two or more models. Specifically, AIC combines the log-likelihood of the predicted probabilities with a penalty for the number of estimated parameters. Lower AIC values indicate a more favorable balance between model complexity and goodness of fit.

## 2.3 e) + f) - Ming

# 3 Results

## 3.1 a) - Sujan

## 3.2 b) - Angel

As previously mentioned, a key assumption of logistic regression is that there is no severe multicollinearity between predictors. We attempted to test whether our data violated this assumption by creating a pairwise Pearson coefficient matrix that included all our predictors.

Table 1: Pearson correlation matrix among predictors

	PCTBACHMOR	MEDHHINC	FATAL_OR_M	OVERTURNED	CELL_PHONE	SPEEDING	AGGRESSIVE	DRIVER1617	DRIVER65PLUS
PCTBACHMOR	1.000	0.478	-0.015	0.009	-0.001	-0.001	0.027	-0.003	0.026
MEDHHINC	0.478	1.000	-0.018	0.028	0.002	0.012	0.043	0.023	0.050
FATAL_OR_M	-0.015	-0.018	1.000	0.033	0.002	0.082	-0.011	-0.003	-0.013
OVERTURNED	0.009	0.028	0.033	1.000	-0.001	0.059	0.016	0.004	-0.020
CELL_PHONE	-0.001	0.002	0.002	-0.001	1.000	-0.004	-0.026	0.001	-0.003
SPEEDING	-0.001	0.012	0.082	0.059	-0.004	1.000	0.212	0.016	-0.033
AGGRESSIVE	0.027	0.043	-0.011	0.016	-0.026	0.212	1.000	0.028	0.015
DRIVER1617	-0.003	0.023	-0.003	0.004	0.001	0.016	0.028	1.000	-0.021
DRIVER65PLUS	0.026	0.050	-0.013	-0.020	-0.003	-0.033	0.015	-0.021	1.000

Table (#) presents the Pearson correlation coefficients between all binary and continuous predictors. Pearson coefficients,  $r$ , range from 1 to -1 and can be interpreted as 1 indicating strong positive linear correlation, -1 indicating strong negative linear correlation, and 0 indicating no linear correlation. Because correlation coefficients are rarely perfectly negative or positive, the threshold considered to indicate moderate correlation is an absolute value of  $r$  ( $|r|$ ) between 0.5 and 0.8 while  $|r| < 0.5$  indicates weak correlation and  $|r| > 0.8$  indicates strong correlation.

The coefficients that include binary variables are all uniformly small, near-zero values. Because the Pearson correlation coefficient is designed to measure the strength of a linear relationship between continuous variables, it is not an ideal measure of association when applied to binary predictors. As a result, Pearson correlation is not as accurate for assessing relationships between binary predictors (or between binary and continuous predictors) and may misrepresent the true association. This means that in our analysis, while the Pearson coefficients are uniformly small, we must be cautious or explore an alternative method in interpreting them.

The correlation coefficient between the two continuous variables, PCTBACHMOR and MEDHHINC, is, as expected, the highest as  $r=0.478$ . Multicollinearity is considered to occur

when two or more predictors are very strongly correlated ( $|r| > 0.9$ ). The correlation between PCTBACHMOR and MEDHHINC is far below this threshold, suggesting no severe multicollinearity.

### **3.3 c) - Ming**

## **4 Discussion**