Analyses

Martijn Straatsburg

Generation date: Jun 08, 2025 - 02:14:33

## Packages

suppressPackageStartupMessages(library(ggplot2))  
suppressPackageStartupMessages(library(jsonlite))  
suppressPackageStartupMessages(library(vcd))  
suppressPackageStartupMessages(library(vcdExtra))  
suppressPackageStartupMessages(library(DescTools))  
suppressPackageStartupMessages(library(car))  
suppressPackageStartupMessages(library(pROC))  
suppressPackageStartupMessages(library(ResourceSelection))  
suppressPackageStartupMessages(library(tidyverse))  
suppressPackageStartupMessages(library(dplyr))  
suppressPackageStartupMessages(library(reshape2))  
suppressPackageStartupMessages(library(effects))  
suppressPackageStartupMessages(library(corrplot))  
suppressPackageStartupMessages(library(logistf))  
suppressPackageStartupMessages(library(RVAideMemoire))  
suppressPackageStartupMessages(library(rstatix))

## Data

df <- fromJSON("predicted-dataset-updated.json")  
df <- df %>%  
 mutate(  
 persuasion\_success = factor(persuasion\_success,  
 levels = c(0 , 1),  
 labels = c("No Delta", "Yes Delta")),  
 story\_class = factor(story\_class),  
 suspense = as.integer(suspense),  
 curiosity = as.integer(curiosity),  
 surprise = as.integer(surprise),  
 level\_suspense = factor(level\_suspense,  
 levels = c("low", "medium", "high")),  
 level\_curiosity = factor(level\_curiosity,  
 levels = c("low", "medium", "high")),  
 level\_surprise = factor(level\_surprise,  
 levels = c("low", "medium", "high")),  
 binary\_suspense = factor(binary\_suspense,  
 levels = c("under", "over")),  
 binary\_curiosity = factor(binary\_curiosity,  
 levels = c("under", "over")),  
 binary\_surprise = factor(binary\_surprise,  
 levels = c("under", "over"))  
 )  
#str(df)

## NEW ANALYSES

#–– 1. Story Class––  
table\_story <- table(df$story\_class, df$persuasion\_success)  
chisq\_story <- chisq.test(table\_story)  
cramerv\_story <- assocstats(table\_story)$cramer  
cramerv\_ci <- CramerV(table\_story, conf.level = 0.95)  
  
knitr::kable(  
 cbind(  
 Count = as.vector(table\_story),  
 Expected = as.vector(chisq\_story$expected),  
 `Prop` = round(prop.table(table\_story, 1) \* 100, 2)  
 ),  
 caption = "Story Class × Persuasion Success"  
)

## Warning in cbind(Count = as.vector(table\_story), Expected =  
## as.vector(chisq\_story$expected), : number of rows of result is not a multiple  
## of vector length (arg 1)

Story Class × Persuasion Success

|  | Count | Expected | No Delta | Yes Delta |
| --- | --- | --- | --- | --- |
| Not Story | 82938 | 82775.16 | 98.61 | 1.39 |
| Story | 13085 | 13247.84 | 97.21 | 2.79 |

cat(  
 "\nPearson χ² =", round(chisq\_story$statistic, 3),  
 "df =", chisq\_story$parameter,  
 "p =", signif(chisq\_story$p.value, 3), "\n",  
 "Cramer’s V =", round(cramerv\_story, 3),  
 "95% CI [", round(cramerv\_ci["lwr.ci"], 3), ",", round(cramerv\_ci["upr.ci"], 3), "]\n"  
)

##   
## Pearson χ² = 145.74 df = 1 p = 1.48e-33   
## Cramer’s V = 0.039 95% CI [ 0.032 , 0.045 ]

#–– 2. Binary Suspense ––  
table\_bin\_susp <- table(df$binary\_suspense, df$persuasion\_success)  
chi\_bin\_susp <- chisq.test(table\_bin\_susp)  
cramerv\_susp <- assocstats(table\_bin\_susp)$cramer  
cramerv\_ci\_s <- CramerV(table\_bin\_susp, conf.level = 0.95)  
  
knitr::kable(  
 cbind(  
 Count = as.vector(table\_bin\_susp),  
 Expected = as.vector(chi\_bin\_susp$expected),  
 `Prop` = round(prop.table(table\_bin\_susp, 1) \* 100, 2)  
 ),  
 caption = "Binary Suspense (Under/Over 2.5) × Persuasion Success"  
)

## Warning in cbind(Count = as.vector(table\_bin\_susp), Expected =  
## as.vector(chi\_bin\_susp$expected), : number of rows of result is not a multiple  
## of vector length (arg 1)

Binary Suspense (Under/Over 2.5) × Persuasion Success

|  | Count | Expected | No Delta | Yes Delta |
| --- | --- | --- | --- | --- |
| under | 89004 | 88916.345 | 98.51 | 1.49 |
| over | 7019 | 7106.655 | 97.20 | 2.80 |

cat(  
 "\nPearson χ² =", round(chi\_bin\_susp$statistic, 3),  
 "df =", chi\_bin\_susp$parameter,  
 "p =", signif(chi\_bin\_susp$p.value, 3), "\n",  
 "Cramer’s V =", round(cramerv\_susp, 3),  
 "95% CI [", round(cramerv\_ci\_s["lwr.ci"], 3), ",", round(cramerv\_ci\_s["upr.ci"], 3), "]\n"  
)

##   
## Pearson χ² = 72.893 df = 1 p = 1.37e-17   
## Cramer’s V = 0.027 95% CI [ 0.021 , 0.034 ]

#–– 3. Binary Curiosity ––  
table\_bin\_cur <- table(df$binary\_curiosity, df$persuasion\_success)  
chi\_bin\_cur <- chisq.test(table\_bin\_cur)  
cramerv\_cur <- assocstats(table\_bin\_cur)$cramer  
cramerv\_ci\_cur <- CramerV(table\_bin\_cur, conf.level = 0.95)  
  
knitr::kable(  
 cbind(  
 Count = as.vector(table\_bin\_cur),  
 Expected = as.vector(chi\_bin\_cur$expected),  
 `Prop` = round(prop.table(table\_bin\_cur, 1) \* 100, 2)  
 ),  
 caption = "Binary Curiosity (Under/Over 2.5) × Persuasion Success"  
)

## Warning in cbind(Count = as.vector(table\_bin\_cur), Expected =  
## as.vector(chi\_bin\_cur$expected), : number of rows of result is not a multiple  
## of vector length (arg 1)

Binary Curiosity (Under/Over 2.5) × Persuasion Success

|  | Count | Expected | No Delta | Yes Delta |
| --- | --- | --- | --- | --- |
| under | 77427 | 77254.98 | 98.64 | 1.36 |
| over | 18596 | 18768.02 | 97.51 | 2.49 |

cat(  
 "\nPearson χ² =", round(chi\_bin\_cur$statistic, 3),  
 "df =", chi\_bin\_cur$parameter,  
 "p =", signif(chi\_bin\_cur$p.value, 3), "\n",  
 "Cramer’s V =", round(cramerv\_cur, 3),  
 "95% CI [", round(cramerv\_ci\_cur["lwr.ci"], 3), ",", round(cramerv\_ci\_cur["upr.ci"], 3), "]\n"  
)

##   
## Pearson χ² = 123.044 df = 1 p = 1.36e-28   
## Cramer’s V = 0.036 95% CI [ 0.029 , 0.042 ]

#–– 4. Binary Surprise ––  
table\_bin\_sur <- table(df$binary\_surprise, df$persuasion\_success)  
chi\_bin\_sur <- chisq.test(table\_bin\_sur)  
cramerv\_sur <- assocstats(table\_bin\_sur)$cramer  
cramerv\_ci\_sur <- CramerV(table\_bin\_sur, conf.level = 0.95)  
  
knitr::kable(  
 cbind(  
 Count = as.vector(table\_bin\_sur),  
 Expected = as.vector(chi\_bin\_sur$expected),  
 `Prop` = round(prop.table(table\_bin\_sur, 1) \* 100, 2)  
 ),  
 caption = "Binary Surprise (Under/Over 2.5) × Persuasion Success"  
)

## Warning in cbind(Count = as.vector(table\_bin\_sur), Expected =  
## as.vector(chi\_bin\_sur$expected), : number of rows of result is not a multiple  
## of vector length (arg 1)

Binary Surprise (Under/Over 2.5) × Persuasion Success

|  | Count | Expected | No Delta | Yes Delta |
| --- | --- | --- | --- | --- |
| under | 87416 | 87349.555 | 98.49 | 1.51 |
| over | 8607 | 8673.445 | 97.66 | 2.34 |

cat(  
 "\nPearson χ² =", round(chi\_bin\_sur$statistic, 3),  
 "df =", chi\_bin\_sur$parameter,  
 "p =", signif(chi\_bin\_sur$p.value, 3), "\n",  
 "Cramer’s V =", round(cramerv\_sur, 3),  
 "95% CI [", round(cramerv\_ci\_sur["lwr.ci"], 3), ",", round(cramerv\_ci\_sur["upr.ci"], 3), "]\n"  
)

##   
## Pearson χ² = 34.807 df = 1 p = 3.64e-09   
## Cramer’s V = 0.019 95% CI [ 0.013 , 0.025 ]

#–– 6. Logistic Regression ––  
model\_bin <- glm(  
 persuasion\_success ~ story\_class +  
 binary\_suspense +  
 binary\_curiosity +  
 binary\_surprise,  
 data = df, family = binomial  
)  
summary(model\_bin)

##   
## Call:  
## glm(formula = persuasion\_success ~ story\_class + binary\_suspense +   
## binary\_curiosity + binary\_surprise, family = binomial, data = df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.29466 0.03099 -138.586 < 2e-16 \*\*\*  
## story\_classStory 0.46297 0.10405 4.450 8.60e-06 \*\*\*  
## binary\_suspenseover 0.13938 0.10733 1.299 0.1941   
## binary\_curiosityover 0.34706 0.07984 4.347 1.38e-05 \*\*\*  
## binary\_surpriseover -0.18474 0.09736 -1.898 0.0577 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 15875 on 97567 degrees of freedom  
## Residual deviance: 15731 on 97563 degrees of freedom  
## AIC: 15741  
##   
## Number of Fisher Scoring iterations: 7

ORs <- exp(coef(model\_bin))  
CIs <- exp(confint(model\_bin))

## Waiting for profiling to be done...

knitr::kable(  
 cbind(OR=ORs, CI\_low=CIs[,1], CI\_high=CIs[,2]),  
 digits=3,  
 caption="Odds Ratios (Binary Predictors) with 95% CI"  
)

Odds Ratios (Binary Predictors) with 95% CI

|  | OR | CI\_low | CI\_high |
| --- | --- | --- | --- |
| (Intercept) | 0.014 | 0.013 | 0.014 |
| story\_classStory | 1.589 | 1.294 | 1.947 |
| binary\_suspenseover | 1.150 | 0.931 | 1.419 |
| binary\_curiosityover | 1.415 | 1.208 | 1.652 |
| binary\_surpriseover | 0.831 | 0.686 | 1.005 |

#–– 8. Interaction Models––  
model\_base <- glm(  
 persuasion\_success ~ story\_class +  
 binary\_suspense +  
 binary\_curiosity +  
 binary\_surprise,  
 data = df, family = binomial  
)  
  
model\_inter1 <- update(  
 model\_base,  
 . ~ . + story\_class:binary\_suspense + story\_class:binary\_curiosity + story\_class:binary\_surprise  
)  
  
model\_inter2 <- glm(  
 persuasion\_success ~ story\_class \*  
 (binary\_suspense + binary\_curiosity + binary\_surprise),  
 data = df, family = binomial  
)  
  
summary(model\_inter1)

##   
## Call:  
## glm(formula = persuasion\_success ~ story\_class + binary\_suspense +   
## binary\_curiosity + binary\_surprise + story\_class:binary\_suspense +   
## story\_class:binary\_curiosity + story\_class:binary\_surprise,   
## family = binomial, data = df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.29045 0.03144 -136.458 < 2e-16 \*\*\*  
## story\_classStory 0.38258 0.17969 2.129 0.033243 \*   
## binary\_suspenseover 0.54013 0.43012 1.256 0.209201   
## binary\_curiosityover 0.35177 0.09310 3.778 0.000158 \*\*\*  
## binary\_surpriseover -0.50651 0.23459 -2.159 0.030841 \*   
## story\_classStory:binary\_suspenseover -0.43289 0.44434 -0.974 0.329938   
## story\_classStory:binary\_curiosityover 0.05749 0.19303 0.298 0.765832   
## story\_classStory:binary\_surpriseover 0.39985 0.25927 1.542 0.123015   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 15875 on 97567 degrees of freedom  
## Residual deviance: 15728 on 97560 degrees of freedom  
## AIC: 15744  
##   
## Number of Fisher Scoring iterations: 7

summary(model\_inter2)

##   
## Call:  
## glm(formula = persuasion\_success ~ story\_class \* (binary\_suspense +   
## binary\_curiosity + binary\_surprise), family = binomial, data = df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.29045 0.03144 -136.458 < 2e-16 \*\*\*  
## story\_classStory 0.38258 0.17969 2.129 0.033243 \*   
## binary\_suspenseover 0.54013 0.43012 1.256 0.209201   
## binary\_curiosityover 0.35177 0.09310 3.778 0.000158 \*\*\*  
## binary\_surpriseover -0.50651 0.23459 -2.159 0.030841 \*   
## story\_classStory:binary\_suspenseover -0.43289 0.44434 -0.974 0.329938   
## story\_classStory:binary\_curiosityover 0.05749 0.19303 0.298 0.765832   
## story\_classStory:binary\_surpriseover 0.39985 0.25927 1.542 0.123015   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 15875 on 97567 degrees of freedom  
## Residual deviance: 15728 on 97560 degrees of freedom  
## AIC: 15744  
##   
## Number of Fisher Scoring iterations: 7

ORs\_i1 <- exp(coef(model\_inter1))  
CIs\_i1 <- exp(confint(model\_inter1))

## Waiting for profiling to be done...

knitr::kable(  
 cbind(OR = ORs\_i1, CI\_low = CIs\_i1[,1], CI\_high = CIs\_i1[,2]),  
 digits = 3,  
 caption = "Odds Ratios & 95% CI, Interaction Model 1"  
)

Odds Ratios & 95% CI, Interaction Model 1

|  | OR | CI\_low | CI\_high |
| --- | --- | --- | --- |
| (Intercept) | 0.014 | 0.013 | 0.015 |
| story\_classStory | 1.466 | 1.017 | 2.060 |
| binary\_suspenseover | 1.716 | 0.660 | 3.668 |
| binary\_curiosityover | 1.422 | 1.180 | 1.700 |
| binary\_surpriseover | 0.603 | 0.369 | 0.930 |
| story\_classStory:binary\_suspenseover | 0.649 | 0.293 | 1.724 |
| story\_classStory:binary\_curiosityover | 1.059 | 0.733 | 1.564 |
| story\_classStory:binary\_surpriseover | 1.492 | 0.916 | 2.542 |

ORs\_i1 <- exp(coef(model\_inter2))  
CIs\_i1 <- exp(confint(model\_inter2))

## Waiting for profiling to be done...

knitr::kable(  
 cbind(OR = ORs\_i1, CI\_low = CIs\_i1[,1], CI\_high = CIs\_i1[,2]),  
 digits = 3,  
 caption = "Odds Ratios & 95% CI, Interaction Model 1"  
)

Odds Ratios & 95% CI, Interaction Model 1

|  | OR | CI\_low | CI\_high |
| --- | --- | --- | --- |
| (Intercept) | 0.014 | 0.013 | 0.015 |
| story\_classStory | 1.466 | 1.017 | 2.060 |
| binary\_suspenseover | 1.716 | 0.660 | 3.668 |
| binary\_curiosityover | 1.422 | 1.180 | 1.700 |
| binary\_surpriseover | 0.603 | 0.369 | 0.930 |
| story\_classStory:binary\_suspenseover | 0.649 | 0.293 | 1.724 |
| story\_classStory:binary\_curiosityover | 1.059 | 0.733 | 1.564 |
| story\_classStory:binary\_surpriseover | 1.492 | 0.916 | 2.542 |

anova\_tab <- anova(  
 model\_base, model\_inter1, model\_inter2,  
 test = "Chisq"  
)  
knitr::kable(  
 anova\_tab,  
 digits = c(NA, 0, 3, 3, 3),  
 caption = "Likelihood‐Ratio Tests: Base vs. Interaction Models"  
)

Likelihood‐Ratio Tests: Base vs. Interaction Models

| Resid. Df | Resid. Dev | Df | Deviance | Pr(>Chi) |
| --- | --- | --- | --- | --- |
| NA | 15731 | NA | NA | NA |
| NA | 15728 | 3 | 3.087 | 0.378 |
| NA | 15728 | 0 | 0.000 | NA |

## OLD; INCLUDING ALL CODE BLOCKS AFTER THIS ONE

# contingency table  
(table\_story <- table(df$story\_class, df$persuasion\_success))

##   
## No Delta Yes Delta  
## Not Story 82938 1169  
## Story 13085 376

# expected counts  
chisq.test(table\_story)$expected

##   
## No Delta Yes Delta  
## Not Story 82775.16 1331.8436  
## Story 13247.84 213.1564

# counts and row %'s table  
prop.table(table\_story, 1) \* 100

##   
## No Delta Yes Delta  
## Not Story 98.610104 1.389896  
## Story 97.206745 2.793255

# chi-square test  
(chisq\_story <- chisq.test(table\_story))

##   
## Pearson's Chi-squared test with Yates' continuity correction  
##   
## data: table\_story  
## X-squared = 145.74, df = 1, p-value < 2.2e-16

# cramer's v (effect size) with CIs  
cramerv\_story <- assocstats(table\_story)  
cramerv\_story$cramer

## [1] 0.03876783

(cramerv\_ci\_story <- CramerV(table\_story, conf.level = 0.95))

## Cramer V lwr.ci upr.ci   
## 0.03876783 0.03249299 0.04504272

# mosaic plot  
#mosaicplot(table\_story, shade = T, main = "Mosaic: Story vs. Persuasion Success")  
  
# stacked bar chart  
#ggplot(df, aes(x=story\_class, fill=persuasion\_success)) + geom\_bar(position='fill') + scale\_y\_continuous(labels = scales::percent\_format()) + labs(y = 'Percent', x = 'Story Class', fill = 'Delta') + theme\_minimal()

## Suspense (low/medium/high)

(table\_lvl\_suspense <- table(df$level\_suspense, df$persuasion\_success))

##   
## No Delta Yes Delta  
## low 89004 1343  
## medium 6305 179  
## high 714 23

chisq.test(table\_lvl\_suspense)$expected

##   
## No Delta Yes Delta  
## low 88916.3453 1430.65467  
## medium 6381.3251 102.67485  
## high 725.3295 11.67048

(chisq\_lvl\_suspense <- chisq.test(table\_lvl\_suspense))

##   
## Pearson's Chi-squared test  
##   
## data: table\_lvl\_suspense  
## X-squared = 74.283, df = 2, p-value < 2.2e-16

cramerv\_lvl\_suspense <- assocstats(table\_lvl\_suspense)  
  
cramerv\_lvl\_suspense$cramer

## [1] 0.02759249

(gkgamma\_lvl\_suspense <- GKgamma(table\_lvl\_suspense))

## gamma : 0.31   
## std. error : 0.034   
## CI : 0.243 0.377

(catest\_lvl\_suspense <- CochranArmitageTest(x = table(df$level\_suspense, df$persuasion\_success)))

##   
## Cochran-Armitage test for trend  
##   
## data: table(df$level\_suspense, df$persuasion\_success)  
## Z = -8.4606, dim = 3, p-value < 2.2e-16  
## alternative hypothesis: two.sided

#df %>% group\_by(level\_suspense, persuasion\_success) %>% summarise(n = n()) %>% group\_by(level\_suspense) %>% mutate(pct = n / sum(n) \* 100) %>% filter(persuasion\_success == 'Yes Delta') %>% ggplot(aes(x = level\_suspense, y = pct, group=1)) + geom\_line() + geom\_point() + labs(title='Yes-Delta % by Suspense Level', y='Percent Yes', x='Suspense Level') + theme\_minimal()  
  
#resid\_lvl\_suspense <- chisq.test(table\_lvl\_suspense)$stdres  
#res\_df <- melt(resid\_lvl\_suspense)  
#colnames(res\_df) <- c('Level','Delta','StdResid')  
  
#ggplot(res\_df, aes(x=Delta, y=Level, fill=StdResid)) + geom\_tile() + geom\_text(aes(label = round(StdResid, 2))) + scale\_fill\_gradient2(low='blue', mid='white', high='red') + labs(title='Std Residuals: Suspense vs Delta') + theme\_minimal()

## Curiosity (low/medium/high)

(table\_lvl\_curiosity <- table(df$level\_curiosity, df$persuasion\_success))

##   
## No Delta Yes Delta  
## low 77427 1071  
## medium 13852 333  
## high 4744 141

chisq.test(table\_lvl\_curiosity)$expected

##   
## No Delta Yes Delta  
## low 77254.976 1243.02445  
## medium 13960.379 224.62103  
## high 4807.645 77.35451

(chisq\_lvl\_curiosity <- chisq.test(table\_lvl\_curiosity))

##   
## Pearson's Chi-squared test  
##   
## data: table\_lvl\_curiosity  
## X-squared = 130.53, df = 2, p-value < 2.2e-16

cramerv\_lvl\_curiosity <- assocstats(table\_lvl\_curiosity)  
  
cramerv\_lvl\_curiosity$cramer

## [1] 0.03657677

(gkgamma\_lvl\_curiosity <- GKgamma(table\_lvl\_curiosity))

## gamma : 0.285   
## std. error : 0.024   
## CI : 0.238 0.332

(catest\_lvl\_curiosity <- CochranArmitageTest(x = table(df$level\_curiosity, df$persuasion\_success)))

##   
## Cochran-Armitage test for trend  
##   
## data: table(df$level\_curiosity, df$persuasion\_success)  
## Z = -11.314, dim = 3, p-value < 2.2e-16  
## alternative hypothesis: two.sided

#df %>% group\_by(level\_curiosity, persuasion\_success) %>% summarise(n = n()) %>% group\_by(level\_curiosity) %>% mutate(pct = n / sum(n) \* 100) %>% filter(persuasion\_success == 'Yes Delta') %>% ggplot(aes(x = level\_curiosity, y = pct, group=1)) + geom\_line() + geom\_point() + labs(title='Yes-Delta % by curiosity Level', y='Percent Yes', x='curiosity Level') + theme\_minimal()  
  
#resid\_lvl\_curiosity <- chisq.test(table\_lvl\_curiosity)$stdres  
#res\_df <- melt(resid\_lvl\_curiosity)  
#colnames(res\_df) <- c('Level','Delta','StdResid')  
  
#ggplot(res\_df, aes(x=Delta, y=Level, fill=StdResid)) + geom\_tile() + geom\_text(aes(label = round(StdResid, 2))) + scale\_fill\_gradient2(low='blue', mid='white', high='red') + labs(title='Std Residuals: curiosity vs Delta') + theme\_minimal()

## Surprise (low/medium/high)

(table\_lvl\_surprise <- table(df$level\_surprise, df$persuasion\_success))

##   
## No Delta Yes Delta  
## low 87416 1339  
## medium 5762 162  
## high 2845 44

chisq.test(table\_lvl\_surprise)$expected

##   
## No Delta Yes Delta  
## low 87349.555 1405.44518  
## medium 5830.193 93.80719  
## high 2843.252 45.74763

(chisq\_lvl\_surprise <- chisq.test(table\_lvl\_surprise))

##   
## Pearson's Chi-squared test  
##   
## data: table\_lvl\_surprise  
## X-squared = 53.63, df = 2, p-value = 2.262e-12

cramerv\_lvl\_surprise <- assocstats(table\_lvl\_surprise)  
  
cramerv\_lvl\_surprise$cramer

## [1] 0.02344496

(gkgamma\_lvl\_surprise <- GKgamma(table\_lvl\_surprise))

## gamma : 0.207   
## std. error : 0.035   
## CI : 0.139 0.276

(catest\_lvl\_surprise <- CochranArmitageTest(x = table(df$level\_surprise, df$persuasion\_success)))

##   
## Cochran-Armitage test for trend  
##   
## data: table(df$level\_surprise, df$persuasion\_success)  
## Z = -4.0874, dim = 3, p-value = 4.362e-05  
## alternative hypothesis: two.sided

#df %>% group\_by(level\_surprise, persuasion\_success) %>% summarise(n = n()) %>% group\_by(level\_surprise) %>% mutate(pct = n / sum(n) \* 100) %>% filter(persuasion\_success == 'Yes Delta') %>% ggplot(aes(x = level\_surprise, y = pct, group=1)) + geom\_line() + geom\_point() + labs(title='Yes-Delta % by surprise Level', y='Percent Yes', x='surprise Level') + theme\_minimal()  
  
#resid\_lvl\_surprise <- chisq.test(table\_lvl\_surprise)$stdres  
#res\_df <- melt(resid\_lvl\_surprise)  
#colnames(res\_df) <- c('Level','Delta','StdResid')  
  
#ggplot(res\_df, aes(x=Delta, y=Level, fill=StdResid)) + geom\_tile() + geom\_text(aes(label = round(StdResid, 2))) + scale\_fill\_gradient2(low='blue', mid='white', high='red') + labs(title='Std Residuals: surprise vs Delta') + theme\_minimal()

## Suspense, Curiosity, Surprise (1-5)

df\_long <- df %>% select(persuasion\_success, suspense, curiosity, surprise) %>%  
 pivot\_longer(-persuasion\_success, names\_to='metric', values\_to='rating')  
  
#ggplot(df\_long, aes(x=rating, color=persuasion\_success)) + geom\_density() + facet\_wrap(~metric, scales='free') + labs(title='Density by Delta Success') + theme\_minimal()  
  
#ggplot(df\_long, aes(x=persuasion\_success, y=rating)) + geom\_boxplot() + facet\_wrap(~metric) + labs(title='Boxplots by Delta Success') + theme\_minimal()  
  
metrics <- c('suspense','curiosity','surprise')  
res\_mwu <- map\_df(metrics, function(m) {  
 formula <- as.formula(paste(m, '~ persuasion\_success'))  
 test <- wilcox.test(formula, data=df, exact=FALSE)  
 medians <- df %>%  
 group\_by(persuasion\_success) %>%  
 summarize(med=median(.data[[m]]),  
 IQR=IQR(.data[[m]]))  
 data.frame(  
 metric = m,  
 U = test$statistic,  
 p = test$p.value,  
 median\_No = medians$med[1],  
 IQR\_No = medians$IQR[1],  
 median\_Yes = medians$med[2],  
 IQR\_Yes = medians$IQR[2]  
 )  
})  
res\_mwu

## metric U p median\_No IQR\_No median\_Yes IQR\_Yes  
## W...1 suspense 66419288 2.571957e-21 1 0 1 1  
## W...2 curiosity 58559253 1.171067e-59 1 1 2 2  
## W...3 surprise 65050689 5.735222e-27 1 1 1 1

## Logistic Regression with Interaction Terms

full\_model <- glm(persuasion\_success ~ story\_class + level\_suspense + level\_curiosity + level\_surprise + story\_class:level\_suspense + story\_class:level\_curiosity + story\_class:level\_surprise, data = df, family = binomial)  
summary(full\_model)

##   
## Call:  
## glm(formula = persuasion\_success ~ story\_class + level\_suspense +   
## level\_curiosity + level\_surprise + story\_class:level\_suspense +   
## story\_class:level\_curiosity + story\_class:level\_surprise,   
## family = binomial, data = df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -4.292934 0.031487 -136.338 < 2e-16  
## story\_classStory 0.404167 0.207662 1.946 0.0516  
## level\_suspensemedium 0.575612 0.445446 1.292 0.1963  
## level\_suspensehigh -6.662651 119.468077 -0.056 0.9555  
## level\_curiositymedium 0.389558 0.092914 4.193 2.76e-05  
## level\_curiosityhigh -0.700543 0.727030 -0.964 0.3353  
## level\_surprisemedium -0.167906 0.250764 -0.670 0.5031  
## level\_surprisehigh -1.701052 0.712551 -2.387 0.0170  
## story\_classStory:level\_suspensemedium -0.446846 0.480952 -0.929 0.3528  
## story\_classStory:level\_suspensehigh 7.076620 119.468326 0.059 0.9528  
## story\_classStory:level\_curiositymedium -0.002725 0.223749 -0.012 0.9903  
## story\_classStory:level\_curiosityhigh 1.051528 0.750831 1.400 0.1614  
## story\_classStory:level\_surprisemedium 0.205176 0.277720 0.739 0.4600  
## story\_classStory:level\_surprisehigh 1.123657 0.736517 1.526 0.1271  
##   
## (Intercept) \*\*\*  
## story\_classStory .   
## level\_suspensemedium   
## level\_suspensehigh   
## level\_curiositymedium \*\*\*  
## level\_curiosityhigh   
## level\_surprisemedium   
## level\_surprisehigh \*   
## story\_classStory:level\_suspensemedium   
## story\_classStory:level\_suspensehigh   
## story\_classStory:level\_curiositymedium   
## story\_classStory:level\_curiosityhigh   
## story\_classStory:level\_surprisemedium   
## story\_classStory:level\_surprisehigh   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 15875 on 97567 degrees of freedom  
## Residual deviance: 15707 on 97554 degrees of freedom  
## AIC: 15735  
##   
## Number of Fisher Scoring iterations: 9

ORs <- exp(coef(full\_model))  
CIs <- exp(confint(full\_model))

## Waiting for profiling to be done...

## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred  
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred

odds\_table <- cbind(OR = ORs, CI\_low = CIs[,1], CI\_high = CIs[,2])  
knitr::kable(odds\_table, digits=3)

|  | OR | CI\_low | CI\_high |
| --- | --- | --- | --- |
| (Intercept) | 0.014 | 0.013 | 0.015 |
| story\_classStory | 1.498 | 0.983 | 2.220 |
| level\_suspensemedium | 1.778 | 0.668 | 3.941 |
| level\_suspensehigh | 0.001 | NA | 47946410.521 |
| level\_curiositymedium | 1.476 | 1.225 | 1.764 |
| level\_curiosityhigh | 0.496 | 0.081 | 1.620 |
| level\_surprisemedium | 0.845 | 0.500 | 1.341 |
| level\_surprisehigh | 0.182 | 0.030 | 0.572 |
| story\_classStory:level\_suspensemedium | 0.640 | 0.266 | 1.801 |
| story\_classStory:level\_suspensehigh | 1183.960 | 0.000 | NA |
| story\_classStory:level\_curiositymedium | 0.997 | 0.649 | 1.562 |
| story\_classStory:level\_curiosityhigh | 2.862 | 0.822 | 18.163 |
| story\_classStory:level\_surprisemedium | 1.228 | 0.730 | 2.177 |
| story\_classStory:level\_surprisehigh | 3.076 | 0.915 | 19.166 |

(tbl\_vif <- vif(full\_model))

## there are higher-order terms (interactions) in this model  
## consider setting type = 'predictor'; see ?vif

## GVIF Df GVIF^(1/(2\*Df))  
## story\_class 1.195412e+01 1 3.457473  
## level\_suspense 9.519744e+06 2 55.546452  
## level\_curiosity 1.407604e+02 2 3.444452  
## level\_surprise 1.879889e+02 2 3.702824  
## story\_class:level\_suspense 1.077044e+07 2 57.287301  
## story\_class:level\_curiosity 5.586118e+02 2 4.861581  
## story\_class:level\_surprise 2.097013e+02 2 3.805400

(hl <- hoslem.test(as.numeric(df$persuasion\_success) - 1, fitted(full\_model), g=10))

## Warning in hoslem.test(as.numeric(df$persuasion\_success) - 1,  
## fitted(full\_model), : The data did not allow for the requested number of bins.

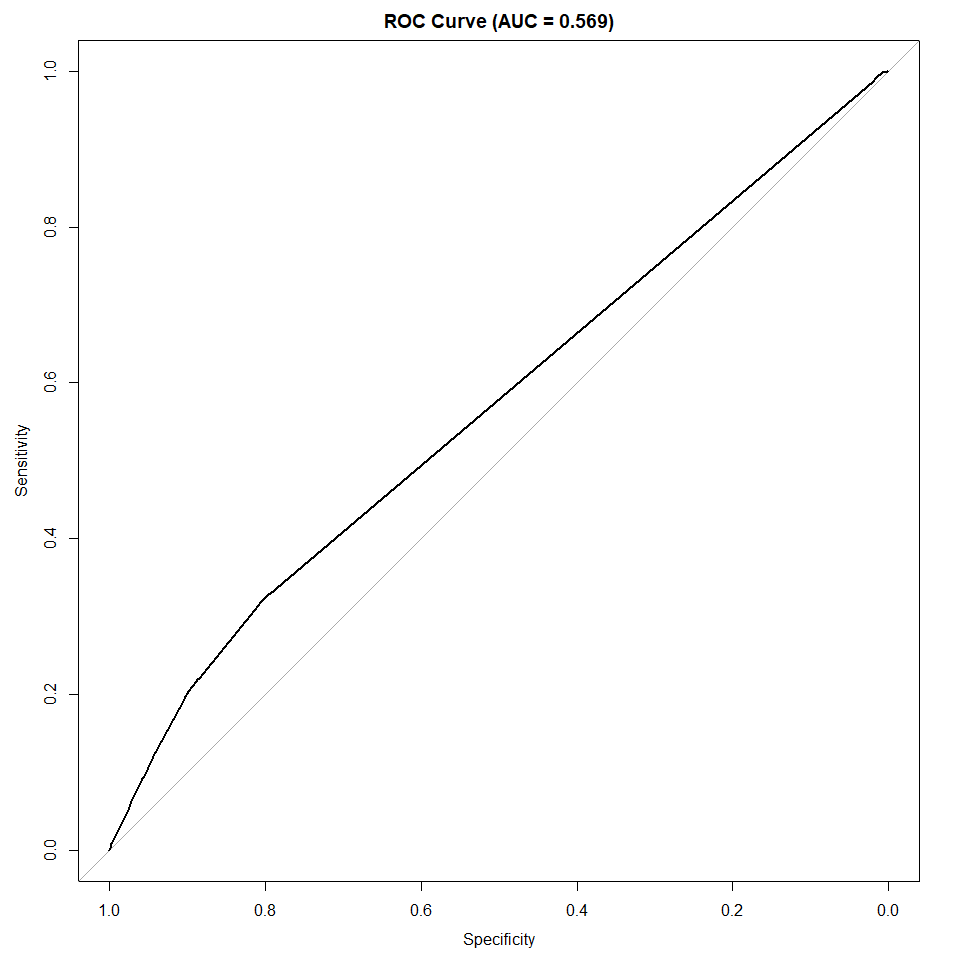
##   
## Hosmer and Lemeshow goodness of fit (GOF) test  
##   
## data: as.numeric(df$persuasion\_success) - 1, fitted(full\_model)  
## X-squared = 0.22963, df = 2, p-value = 0.8915

roc\_obj <- roc(df$persuasion\_success, predict(full\_model, type='response'))

## Setting levels: control = No Delta, case = Yes Delta

## Setting direction: controls < cases

plot(roc\_obj, main = paste0('ROC Curve (AUC = ', round(auc(roc\_obj),3), ')'))



## Old stuff

#(table\_story\_class <- table(df$story\_class, df$persuasion\_success))  
#chisq.test(table\_story\_class)$expected  
#chisq.test(table\_story\_class)  
#assocstats(table\_story\_class)$cramer  
#mosaicplot(table\_story\_class, shade=TRUE, main="Story vs. Persuasion Success")  
# stacked bar chart  
# table with counts and row percentage  
# explain each piece of code and result  
# independence of each post (no repeated measures)  
  
#(table\_level\_suspense <- table(df$level\_suspense, df$persuasion\_success))  
#chisq.test(table\_level\_suspense)$expected  
#chisq.test(table\_level\_suspense)  
#assocstats(table\_level\_suspense)$cramer  
#GKgamma(table\_level\_suspense)  
# ordinal trend test (Cochran-Armitage)  
# profile plot  
# stacked bar chart  
# heatmap  
  
#(table\_level\_curiosity <- table(df$level\_curiosity, df$persuasion\_success))  
#chisq.test(table\_level\_curiosity)$expected  
#chisq.test(table\_level\_curiosity)  
#assocstats(table\_level\_curiosity)$cramer  
#GKgamma(table\_level\_curiosity)  
# ordinal trend test (Cochran-Armitage)  
# profile plot  
# stacked bar chart  
# heatmap  
  
#(table\_level\_surprise <- table(df$level\_surprise, df$persuasion\_success))  
#chisq.test(table\_level\_surprise)$expected  
#chisq.test(table\_level\_surprise)  
#assocstats(table\_level\_surprise)$cramer  
#GKgamma(table\_level\_surprise)  
# ordinal trend test (Cochran-Armitage)  
# profile plot  
# stacked bar chart  
# heatmap  
  
#full\_model <- glm(persuasion\_success ~ story\_class + level\_suspense + level\_curiosity + level\_surprise + story\_class:level\_suspense + story\_class:level\_curiosity + story\_class:level\_surprise, data = df, family = binomial)  
#summary(full\_model)  
#odds\_ratios <- exp(coef(full\_model))  
#conf\_ints <- exp(confint(full\_model))  
#cbind(odds\_ratio = odds\_ratios, low\_conf\_int = conf\_ints[,1], upper\_conf\_int = conf\_ints[,2])

## Everything missing.

# 1. CORRELATION ANALYSIS FOR CONTINUOUS VARIABLES  
  
# Spearman correlations between 1-5 scale variables  
cor\_matrix <- cor(df[,c("suspense", "curiosity", "surprise")],   
 method = "spearman", use = "complete.obs")  
print("Spearman Correlations:")

## [1] "Spearman Correlations:"

print(cor\_matrix)

## suspense curiosity surprise  
## suspense 1.0000000 0.7196841 0.7903028  
## curiosity 0.7196841 1.0000000 0.7854596  
## surprise 0.7903028 0.7854596 1.0000000

# Test significance of correlations (suppress tie warnings - normal with ordinal data)  
cor\_test\_results <- suppressWarnings(cor.test(df$suspense, df$curiosity, method = "spearman"))  
cor\_test\_results2 <- suppressWarnings(cor.test(df$suspense, df$surprise, method = "spearman"))  
cor\_test\_results3 <- suppressWarnings(cor.test(df$curiosity, df$surprise, method = "spearman"))  
  
# Create correlation summary table  
correlation\_summary <- data.frame(  
 Variable\_Pair = c("Suspense-Curiosity", "Suspense-Surprise", "Curiosity-Surprise"),  
 Spearman\_rho = c(cor\_test\_results$estimate, cor\_test\_results2$estimate, cor\_test\_results3$estimate),  
 p\_value = c(cor\_test\_results$p.value, cor\_test\_results2$p.value, cor\_test\_results3$p.value),  
 Significance = c(  
 ifelse(cor\_test\_results$p.value < 0.001, "\*\*\*",  
 ifelse(cor\_test\_results$p.value < 0.01, "\*\*",  
 ifelse(cor\_test\_results$p.value < 0.05, "\*", "ns"))),  
 ifelse(cor\_test\_results2$p.value < 0.001, "\*\*\*",  
 ifelse(cor\_test\_results2$p.value < 0.01, "\*\*",  
 ifelse(cor\_test\_results2$p.value < 0.05, "\*", "ns"))),  
 ifelse(cor\_test\_results3$p.value < 0.001, "\*\*\*",  
 ifelse(cor\_test\_results3$p.value < 0.01, "\*\*",  
 ifelse(cor\_test\_results3$p.value < 0.05, "\*", "ns")))  
 )  
)  
  
print("Correlation Test Results:")

## [1] "Correlation Test Results:"

print(correlation\_summary)

## Variable\_Pair Spearman\_rho p\_value Significance  
## 1 Suspense-Curiosity 0.7196841 0 \*\*\*  
## 2 Suspense-Surprise 0.7903028 0 \*\*\*  
## 3 Curiosity-Surprise 0.7854596 0 \*\*\*

# 2. POST-HOC PAIRWISE COMPARISONS FOR CATEGORICAL VARIABLES  
  
# Function to perform pairwise chi-square tests  
perform\_pairwise\_chisq <- function(data, group\_var, outcome\_var) {  
 levels\_list <- levels(data[[group\_var]])  
 n\_comparisons <- choose(length(levels\_list), 2)  
 results <- data.frame()  
   
 for(i in 1:(length(levels\_list)-1)) {  
 for(j in (i+1):length(levels\_list)) {  
 # Subset data for two levels  
 subset\_data <- data[data[[group\_var]] %in% c(levels\_list[i], levels\_list[j]), ]  
 subset\_data[[group\_var]] <- droplevels(subset\_data[[group\_var]])  
   
 # Create contingency table  
 cont\_table <- table(subset\_data[[group\_var]], subset\_data[[outcome\_var]])  
   
 # Perform chi-square test  
 chisq\_result <- chisq.test(cont\_table)  
   
 # Calculate percentages for each group  
 prop\_table <- prop.table(cont\_table, 1) \* 100  
 yes\_delta\_pct1 <- prop\_table[1, "Yes Delta"]  
 yes\_delta\_pct2 <- prop\_table[2, "Yes Delta"]  
   
 results <- rbind(results, data.frame(  
 Comparison = paste(levels\_list[i], "vs", levels\_list[j]),  
 Chi\_Square = round(chisq\_result$statistic, 3),  
 p\_value = chisq\_result$p.value,  
 p\_adjusted = p.adjust(chisq\_result$p.value, method = "bonferroni", n = n\_comparisons),  
 Pct\_YesDelta\_Group1 = round(yes\_delta\_pct1, 2),  
 Pct\_YesDelta\_Group2 = round(yes\_delta\_pct2, 2),  
 stringsAsFactors = FALSE  
 ))  
 }  
 }  
   
 # Add significance indicators  
 results$Significance <- ifelse(results$p\_adjusted < 0.001, "\*\*\*",  
 ifelse(results$p\_adjusted < 0.01, "\*\*",  
 ifelse(results$p\_adjusted < 0.05, "\*", "ns")))  
   
 return(results)  
}  
  
# Pairwise comparisons for suspense levels  
print("=== Pairwise comparisons for Suspense levels ===")

## [1] "=== Pairwise comparisons for Suspense levels ==="

suspense\_pairwise <- perform\_pairwise\_chisq(df, "level\_suspense", "persuasion\_success")  
print(suspense\_pairwise)

## Comparison Chi\_Square p\_value p\_adjusted  
## X-squared low vs medium 62.663 2.452710e-15 7.358131e-15  
## X-squared1 low vs high 12.134 4.950931e-04 1.485279e-03  
## X-squared2 medium vs high 0.197 6.570883e-01 1.000000e+00  
## Pct\_YesDelta\_Group1 Pct\_YesDelta\_Group2 Significance  
## X-squared 1.49 2.76 \*\*\*  
## X-squared1 1.49 3.12 \*\*  
## X-squared2 2.76 3.12 ns

# Pairwise comparisons for curiosity levels   
print("=== Pairwise comparisons for Curiosity levels ===")

## [1] "=== Pairwise comparisons for Curiosity levels ==="

curiosity\_pairwise <- perform\_pairwise\_chisq(df, "level\_curiosity", "persuasion\_success")  
print(curiosity\_pairwise)

## Comparison Chi\_Square p\_value p\_adjusted  
## X-squared low vs medium 77.185 1.556374e-18 4.669122e-18  
## X-squared1 low vs high 73.315 1.105156e-17 3.315469e-17  
## X-squared2 medium vs high 4.133 4.204851e-02 1.261455e-01  
## Pct\_YesDelta\_Group1 Pct\_YesDelta\_Group2 Significance  
## X-squared 1.36 2.35 \*\*\*  
## X-squared1 1.36 2.89 \*\*\*  
## X-squared2 2.35 2.89 ns

# Pairwise comparisons for surprise levels  
print("=== Pairwise comparisons for Surprise levels ===")

## [1] "=== Pairwise comparisons for Surprise levels ==="

surprise\_pairwise <- perform\_pairwise\_chisq(df, "level\_surprise", "persuasion\_success")  
print(surprise\_pairwise)

## Comparison Chi\_Square p\_value p\_adjusted  
## X-squared low vs medium 52.716 3.854993e-13 1.156498e-12  
## X-squared1 low vs high 0.000 1.000000e+00 1.000000e+00  
## X-squared2 medium vs high 11.963 5.426446e-04 1.627934e-03  
## Pct\_YesDelta\_Group1 Pct\_YesDelta\_Group2 Significance  
## X-squared 1.51 2.73 \*\*\*  
## X-squared1 1.51 1.52 ns  
## X-squared2 2.73 1.52 \*\*

# 3. MODEL COMPARISON AND SELECTION  
  
# Main effects only model  
main\_model <- glm(persuasion\_success ~ story\_class + level\_suspense + level\_curiosity + level\_surprise,  
 family = binomial, data = df)  
  
full\_model <- glm(persuasion\_success ~ story\_class + level\_suspense + level\_curiosity + level\_surprise + story\_class:level\_suspense + story\_class:level\_curiosity + story\_class:level\_surprise, family = binomial, data = df)  
  
# Likelihood ratio test comparing models  
lr\_test <- anova(main\_model, full\_model, test = "LRT")  
print("Likelihood Ratio Test - Main vs Full Model:")

## [1] "Likelihood Ratio Test - Main vs Full Model:"

print(lr\_test)

## Analysis of Deviance Table  
##   
## Model 1: persuasion\_success ~ story\_class + level\_suspense + level\_curiosity +   
## level\_surprise  
## Model 2: persuasion\_success ~ story\_class + level\_suspense + level\_curiosity +   
## level\_surprise + story\_class:level\_suspense + story\_class:level\_curiosity +   
## story\_class:level\_surprise  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)  
## 1 97560 15713   
## 2 97554 15707 6 6.7766 0.342

# AIC/BIC comparison  
model\_comparison <- data.frame(  
 Model = c("Main Effects", "Full Interactions"),  
 AIC = c(AIC(main\_model), AIC(full\_model)),  
 BIC = c(BIC(main\_model), BIC(full\_model)),  
 Deviance = c(deviance(main\_model), deviance(full\_model))  
)  
print("Model Comparison:")

## [1] "Model Comparison:"

print(model\_comparison)

## Model AIC BIC Deviance  
## 1 Main Effects 15729.35 15805.26 15713.35  
## 2 Full Interactions 15734.58 15867.41 15706.58

# 4. ADDRESSING MULTICOLLINEARITY - CENTERED VARIABLES  
  
# Create centered versions of variables for interactions  
df$story\_numeric <- as.numeric(df$story\_class) - 1 # 0/1 coding  
df$suspense\_centered <- scale(as.numeric(df$level\_suspense), center = TRUE, scale = FALSE)[,1]  
df$curiosity\_centered <- scale(as.numeric(df$level\_curiosity), center = TRUE, scale = FALSE)[,1]  
df$surprise\_centered <- scale(as.numeric(df$level\_surprise), center = TRUE, scale = FALSE)[,1]  
  
# Interaction model with centered variables  
centered\_model <- glm(persuasion\_success ~ story\_class + level\_suspense + level\_curiosity + level\_surprise +  
 story\_numeric:suspense\_centered + story\_numeric:curiosity\_centered + story\_numeric:surprise\_centered,  
 family = binomial, data = df)  
  
print("Centered Interaction Model Summary:")

## [1] "Centered Interaction Model Summary:"

summary(centered\_model)

##   
## Call:  
## glm(formula = persuasion\_success ~ story\_class + level\_suspense +   
## level\_curiosity + level\_surprise + story\_numeric:suspense\_centered +   
## story\_numeric:curiosity\_centered + story\_numeric:surprise\_centered,   
## family = binomial, data = df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -4.28968 0.03137 -136.751 < 2e-16 \*\*\*  
## story\_classStory 0.33903 0.15328 2.212 0.02698 \*   
## level\_suspensemedium 0.51230 0.42889 1.194 0.23230   
## level\_suspensehigh 1.00328 0.90345 1.111 0.26678   
## level\_curiositymedium 0.35832 0.09101 3.937 8.25e-05 \*\*\*  
## level\_curiosityhigh 0.04912 0.32169 0.153 0.87865   
## level\_surprisemedium -0.29839 0.20554 -1.452 0.14657   
## level\_surprisehigh -1.32019 0.42002 -3.143 0.00167 \*\*   
## story\_numeric:suspense\_centered -0.29086 0.44626 -0.652 0.51454   
## story\_numeric:curiosity\_centered 0.15307 0.19217 0.797 0.42570   
## story\_numeric:surprise\_centered 0.36411 0.21780 1.672 0.09457 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 15875 on 97567 degrees of freedom  
## Residual deviance: 15709 on 97557 degrees of freedom  
## AIC: 15731  
##   
## Number of Fisher Scoring iterations: 7

# Check VIF for centered model  
print("VIF for Centered Model:")

## [1] "VIF for Centered Model:"

print(vif(centered\_model))

## there are higher-order terms (interactions) in this model  
## consider setting type = 'predictor'; see ?vif

## GVIF Df GVIF^(1/(2\*Df))  
## story\_class 6.511774 1 2.551818  
## level\_suspense 83.123478 2 3.019471  
## level\_curiosity 21.965361 2 2.164884  
## level\_surprise 13.223054 2 1.906922  
## story\_numeric:suspense\_centered 40.310825 1 6.349081  
## story\_numeric:curiosity\_centered 15.931330 1 3.991407  
## story\_numeric:surprise\_centered 11.058611 1 3.325449

# 5. FIRTH'S PENALIZED LIKELIHOOD REGRESSION  
  
# Firth's logistic regression to address separation issues  
firth\_model <- logistf(persuasion\_success ~ story\_class + level\_suspense + level\_curiosity + level\_surprise,  
 data = df)  
  
print("Firth's Penalized Likelihood Regression:")

## [1] "Firth's Penalized Likelihood Regression:"

summary(firth\_model)

## logistf(formula = persuasion\_success ~ story\_class + level\_suspense +   
## level\_curiosity + level\_surprise, data = df)  
##   
## Model fitted by Penalized ML  
## Coefficients:  
## coef se(coef) lower 0.95 upper 0.95 Chisq  
## (Intercept) -4.29517713 0.03111818 -4.35677327 -4.2347682 Inf  
## story\_classStory 0.45426575 0.10492974 0.24771572 0.6598119 18.42788374  
## level\_suspensemedium 0.20099392 0.14861956 -0.09479725 0.4898045 1.78693654  
## level\_suspensehigh 0.50888759 0.23790255 0.01948901 0.9583380 4.13804854  
## level\_curiositymedium 0.36171919 0.08305443 0.19633956 0.5224211 17.76821078  
## level\_curiosityhigh 0.26050513 0.15285836 -0.03825810 0.5632582 2.91600699  
## level\_surprisemedium -0.01545902 0.10456469 -0.22303471 0.1878908 0.02179794  
## level\_surprisehigh -0.70348280 0.17357803 -1.05740030 -0.3735217 18.84002598  
## p method  
## (Intercept) 0.000000e+00 2  
## story\_classStory 1.764569e-05 2  
## level\_suspensemedium 1.812999e-01 2  
## level\_suspensehigh 4.192948e-02 2  
## level\_curiositymedium 2.495176e-05 2  
## level\_curiosityhigh 8.770465e-02 2  
## level\_surprisemedium 8.826259e-01 2  
## level\_surprisehigh 1.421525e-05 2  
##   
## Method: 1-Wald, 2-Profile penalized log-likelihood, 3-None  
##   
## Likelihood ratio test=164.3837 on 7 df, p=0, n=97568  
## Wald test = 25456.91 on 7 df, p = 0

# 6. EFFECT SIZE INTERPRETATION  
  
# Function to interpret Cramer's V effect sizes  
interpret\_cramers\_v <- function(v) {  
 case\_when(  
 v < 0.1 ~ "Negligible",  
 v < 0.3 ~ "Small",  
 v < 0.5 ~ "Medium",  
 TRUE ~ "Large"  
 )  
}  
  
# Create effect size summary table  
effect\_sizes <- data.frame(  
 Variable = c("Story Class", "Suspense Level", "Curiosity Level", "Surprise Level"),  
 Cramers\_V = c(0.03876783, 0.02759249, 0.03657677, 0.02344496),  
 Interpretation = c(  
 interpret\_cramers\_v(0.03876783),  
 interpret\_cramers\_v(0.02759249),  
 interpret\_cramers\_v(0.03657677),  
 interpret\_cramers\_v(0.02344496)  
 )  
)  
  
print("Effect Size Interpretations:")

## [1] "Effect Size Interpretations:"

print(effect\_sizes)

## Variable Cramers\_V Interpretation  
## 1 Story Class 0.03876783 Negligible  
## 2 Suspense Level 0.02759249 Negligible  
## 3 Curiosity Level 0.03657677 Negligible  
## 4 Surprise Level 0.02344496 Negligible

# 7. SIMPLIFIED MODEL WITHOUT PROBLEMATIC INTERACTIONS  
  
# Model without suspense interactions (due to high VIF)  
reduced\_model <- glm(persuasion\_success ~ story\_class + level\_suspense + level\_curiosity + level\_surprise +  
 story\_class:level\_curiosity + story\_class:level\_surprise,  
 family = binomial, data = df)  
  
print("Reduced Model (without suspense interactions):")

## [1] "Reduced Model (without suspense interactions):"

summary(reduced\_model)

##   
## Call:  
## glm(formula = persuasion\_success ~ story\_class + level\_suspense +   
## level\_curiosity + level\_surprise + story\_class:level\_curiosity +   
## story\_class:level\_surprise, family = binomial, data = df)  
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -4.29208 0.03146 -136.426 < 2e-16  
## story\_classStory 0.36557 0.20460 1.787 0.0740  
## level\_suspensemedium 0.18477 0.17082 1.082 0.2794  
## level\_suspensehigh 0.42790 0.24329 1.759 0.0786  
## level\_curiositymedium 0.38662 0.09296 4.159 3.2e-05  
## level\_curiosityhigh -0.61775 0.72029 -0.858 0.3911  
## level\_surprisemedium -0.10661 0.23885 -0.446 0.6554  
## level\_surprisehigh -1.69014 0.70966 -2.382 0.0172  
## story\_classStory:level\_curiositymedium 0.03264 0.22113 0.148 0.8827  
## story\_classStory:level\_curiosityhigh 0.95487 0.74231 1.286 0.1983  
## story\_classStory:level\_surprisemedium 0.14334 0.26681 0.537 0.5911  
## story\_classStory:level\_surprisehigh 1.10775 0.73360 1.510 0.1310  
##   
## (Intercept) \*\*\*  
## story\_classStory .   
## level\_suspensemedium   
## level\_suspensehigh .   
## level\_curiositymedium \*\*\*  
## level\_curiosityhigh   
## level\_surprisemedium   
## level\_surprisehigh \*   
## story\_classStory:level\_curiositymedium   
## story\_classStory:level\_curiosityhigh   
## story\_classStory:level\_surprisemedium   
## story\_classStory:level\_surprisehigh   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 15875 on 97567 degrees of freedom  
## Residual deviance: 15707 on 97556 degrees of freedom  
## AIC: 15731  
##   
## Number of Fisher Scoring iterations: 7

print("VIF for Reduced Model:")

## [1] "VIF for Reduced Model:"

print(vif(reduced\_model))

## there are higher-order terms (interactions) in this model  
## consider setting type = 'predictor'; see ?vif

## GVIF Df GVIF^(1/(2\*Df))  
## story\_class 11.603535 1 3.406396  
## level\_suspense 5.582612 2 1.537126  
## level\_curiosity 138.229246 2 3.428862  
## level\_surprise 169.262366 2 3.606950  
## story\_class:level\_curiosity 528.752528 2 4.795271  
## story\_class:level\_surprise 192.097048 2 3.722890

# Compare reduced model to full model  
lr\_test\_reduced <- anova(reduced\_model, full\_model, test = "LRT")  
print("LR Test - Reduced vs Full Model:")

## [1] "LR Test - Reduced vs Full Model:"

print(lr\_test\_reduced)

## Analysis of Deviance Table  
##   
## Model 1: persuasion\_success ~ story\_class + level\_suspense + level\_curiosity +   
## level\_surprise + story\_class:level\_curiosity + story\_class:level\_surprise  
## Model 2: persuasion\_success ~ story\_class + level\_suspense + level\_curiosity +   
## level\_surprise + story\_class:level\_suspense + story\_class:level\_curiosity +   
## story\_class:level\_surprise  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)  
## 1 97556 15707   
## 2 97554 15707 2 0.85836 0.651

# 8. ADDITIONAL MODEL DIAGNOSTICS  
  
# Pseudo R-squared for models  
pseudo\_r2 <- function(model) {  
 1 - (model$deviance / model$null.deviance)  
}  
  
print("Pseudo R-squared values:")

## [1] "Pseudo R-squared values:"

print(paste("Main Model:", round(pseudo\_r2(main\_model), 4)))

## [1] "Main Model: 0.0102"

print(paste("Full Model:", round(pseudo\_r2(full\_model), 4)))

## [1] "Full Model: 0.0106"

print(paste("Reduced Model:", round(pseudo\_r2(reduced\_model), 4)))

## [1] "Reduced Model: 0.0106"

# Create summary table for all models  
model\_summary\_table <- data.frame(  
 Model = c("Main Effects", "Full Interactions", "Reduced Interactions"),  
 AIC = c(AIC(main\_model), AIC(full\_model), AIC(reduced\_model)),  
 BIC = c(BIC(main\_model), BIC(full\_model), BIC(reduced\_model)),  
 Pseudo\_R2 = c(pseudo\_r2(main\_model), pseudo\_r2(full\_model), pseudo\_r2(reduced\_model)),  
 Max\_VIF = c(max(vif(main\_model)), max(vif(full\_model)[,3]), max(vif(reduced\_model)[,3]))  
)

## there are higher-order terms (interactions) in this model  
## consider setting type = 'predictor'; see ?vif  
## there are higher-order terms (interactions) in this model  
## consider setting type = 'predictor'; see ?vif

print("Complete Model Comparison Table:")

## [1] "Complete Model Comparison Table:"

print(model\_summary\_table)

## Model AIC BIC Pseudo\_R2 Max\_VIF  
## 1 Main Effects 15729.35 15805.26 0.01018734 4.691517  
## 2 Full Interactions 15734.58 15867.41 0.01061421 57.287301  
## 3 Reduced Interactions 15731.44 15845.30 0.01056014 4.795271