Analyses

Martijn Straatsburg

Generation date: Jun 08, 2025 - 02:20:55

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)
```

Table 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 Chi-Square =", 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 Chi-Square = 145.74 df = 1 p = 1.48e-33
##  Cramer's V = 0.039 95% CI [ 0.032 , 0.045 ]
```

```r
#-- 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)
```

Table 2: 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

```r
cat(
  "\nPearson Chi-Square =", 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 Chi-Square = 72.893 df = 1 p = 1.37e-17
##  Cramer's V = 0.027 95% CI [ 0.021 , 0.034 ]
```

```r
#-- 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)
```

Table 3: 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 Chi-Square =", 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 Chi-Square = 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)
```

Table 4: 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

```r
cat(
  "\nPearson Chi-Square =", 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 Chi-Square = 34.807 df = 1 p = 3.64e-09
##  Cramer's V = 0.019 95% CI [ 0.013 , 0.025 ]
```

```r
#-- 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
```

```r
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"
)
```

Table 5: 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"
)
```

Table 6: 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"
)
```

Table 7: 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"
)
```

Table 8: 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
```

```r
(cramerv_ci_story <- CramerV(table_story, conf.level = 0.95))
```

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

```r
# 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_continuo
```

Suspense (low/medium/high)

```r
(table_lvl_suspense <- table(df$level_suspense, df$persuasion_success))
```

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

```r
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
```

```r
(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
```

```r
cramerv_lvl_suspense <- assocstats(table_lvl_suspense)

cramerv_lvl_suspense$cramer
```

```
## [1] 0.02759249
```

```r
(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

#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(StdRe
```

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_curiosi

#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(StdRe
```

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

#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(StdRe
```

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

#ggplot(df_long, aes(x=persuasion_success, y=rating)) + geom_boxplot() + facet_wrap(~metric) + labs(tit

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 +
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

	OR	CI_low	CI_high
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.
```

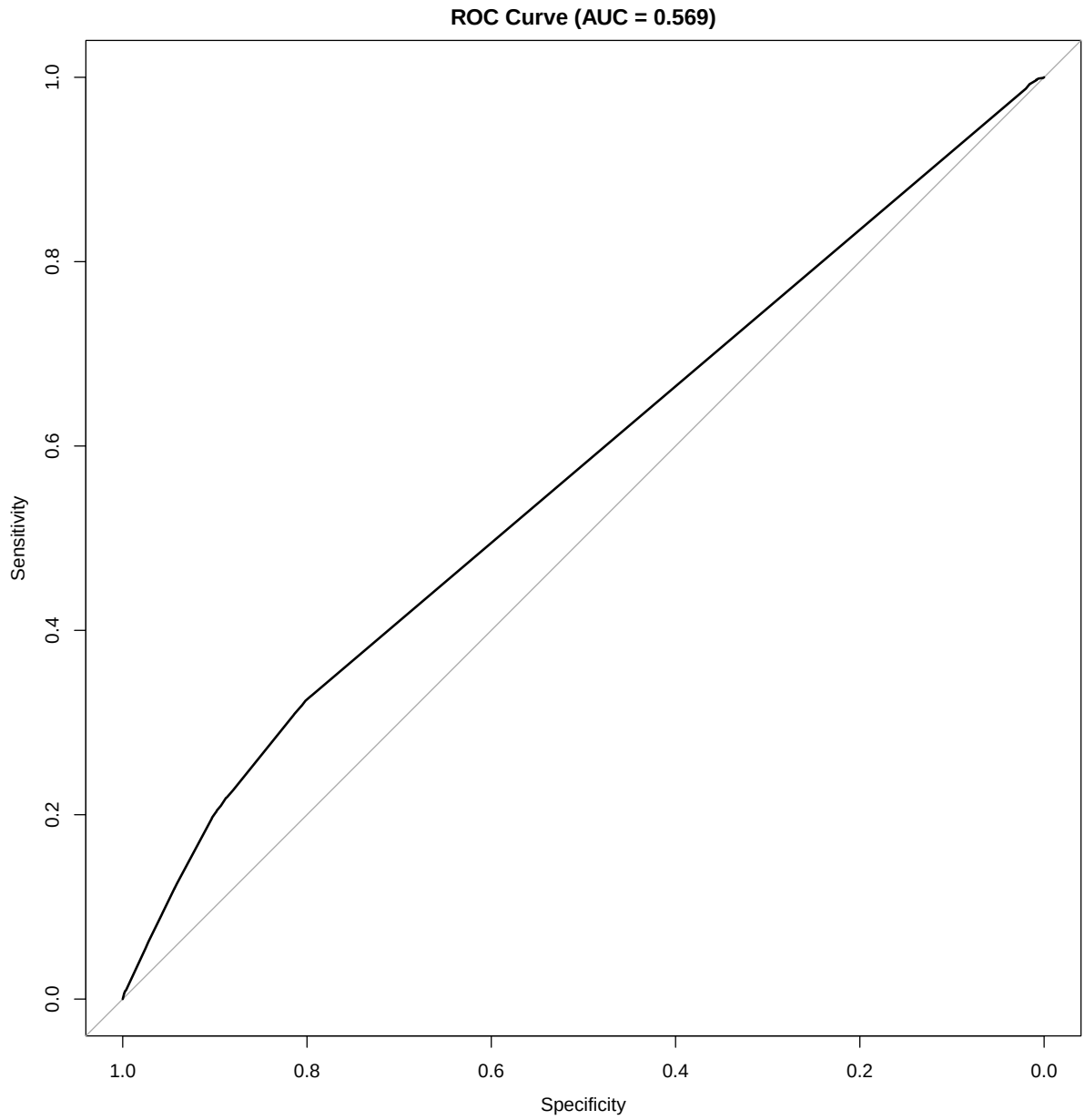
```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  as.numeric(df$persuasion_success) - 1, fitted(full_model)
## X-squared = 0.22963, df = 2, p-value = 0.8915
```

```
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
#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
```

```r
# 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:"
```

```r
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          ***
```

```r
# 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 ==="
```

```r
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
```

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

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

```r
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           **
```

```r
# 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

# 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:"
```

```r
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_surpri
                      story_numeric:suspense_centered + story_numeric:curiosity_centered + story_numeri
                      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_surpr
                       data = df)

print("Firth's Penalized Likelihood Regression:")
```

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

```r
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
```

```r
# 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_surpris
                     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
```