

Analyses

Martijn Straatsburg

Generation date: May 26, 2025 - 16:28:39

Packages

```
library(ggplot2)
library(jsonlite)
suppressPackageStartupMessages(library(vcd))
suppressPackageStartupMessages(library(vcdExtra))
library(DescTools)
suppressPackageStartupMessages(library(car))
suppressPackageStartupMessages(library(pROC))
suppressPackageStartupMessages(library(ResourceSelection))
suppressPackageStartupMessages(library(tidyverse))
suppressPackageStartupMessages(library(dplyr))
suppressPackageStartupMessages(library(reshape2))
library(effects)
```

```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

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)

```

Story

```

# 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

```

```

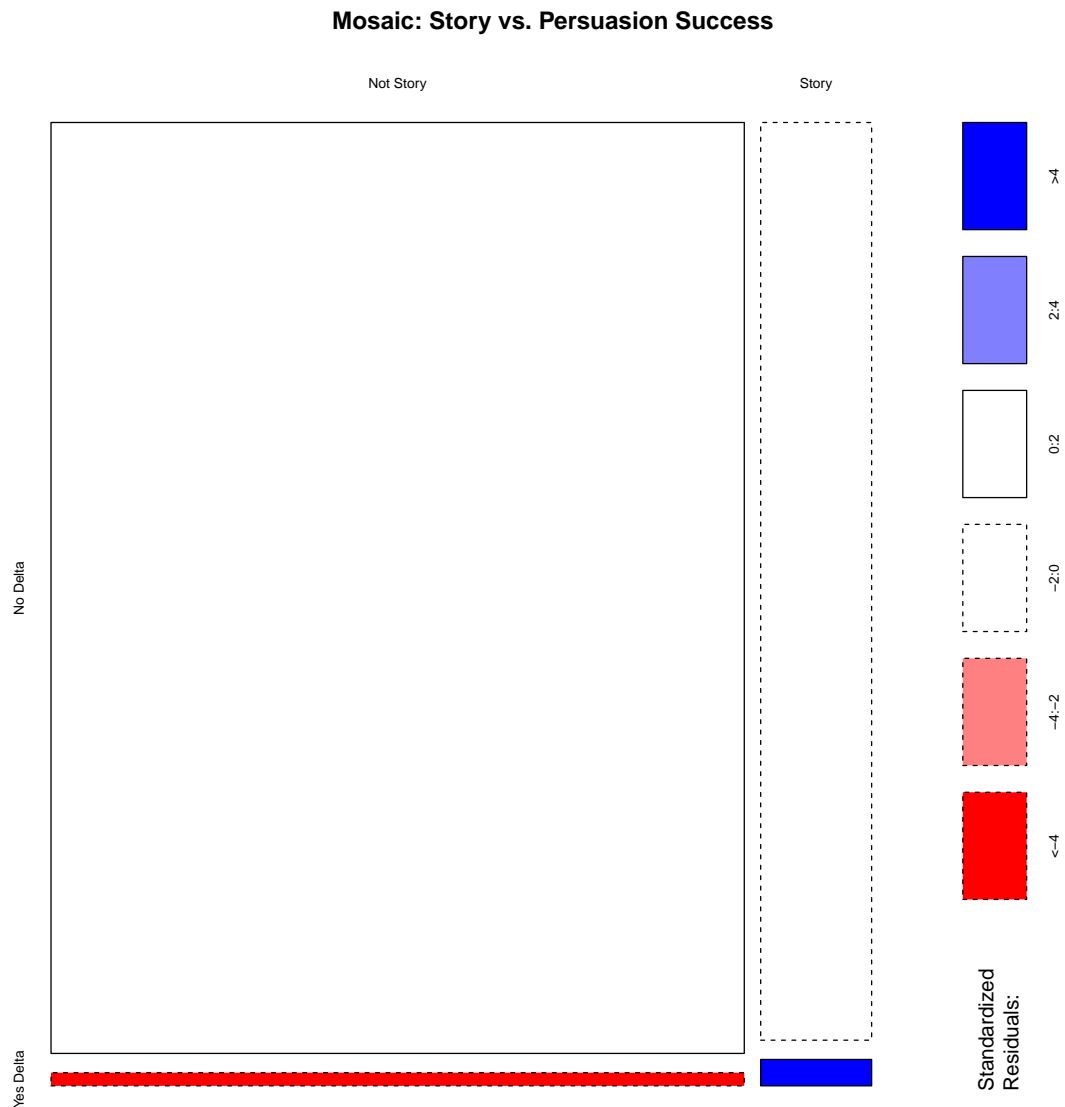
## [1] 0.03876783

```

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

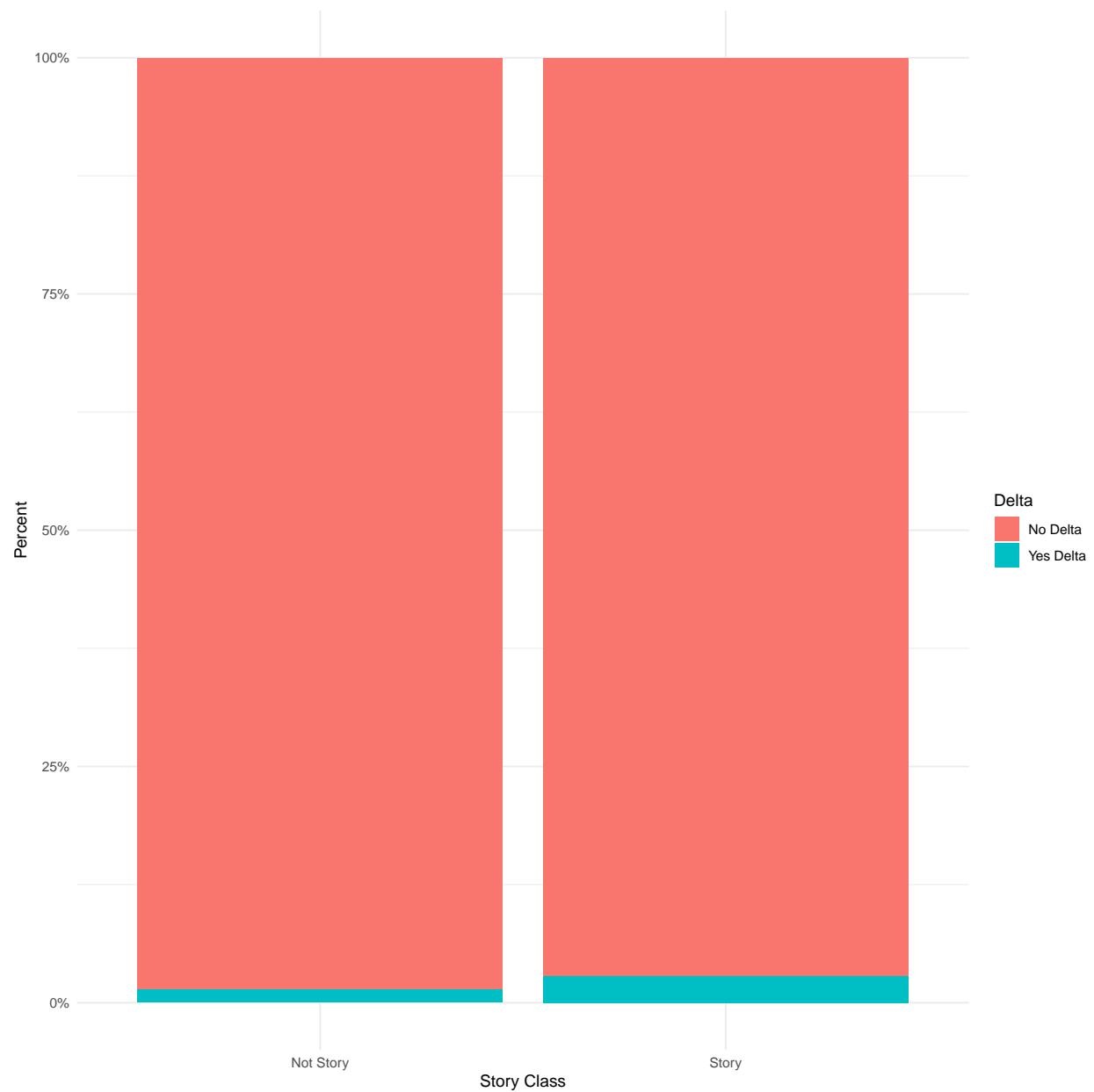
```
##   Cramer V      lwr.ci      upr.ci
## 0.03876783 0.03249299 0.04504272
```

```
# 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))
```

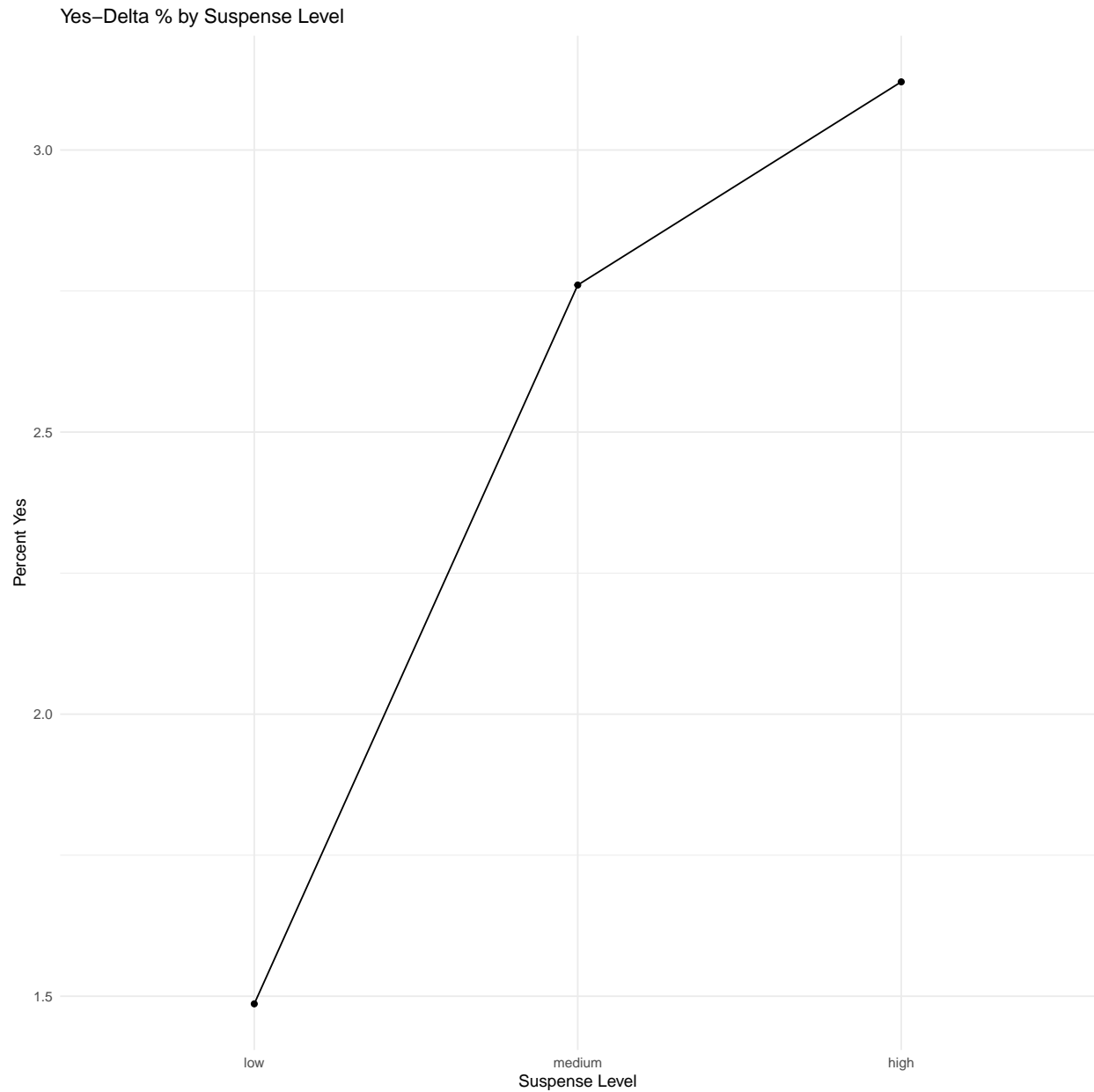
```
## gamma      : 0.31
## std. error  : 0.034
## CI          : 0.243 0.377
```

```
(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
```

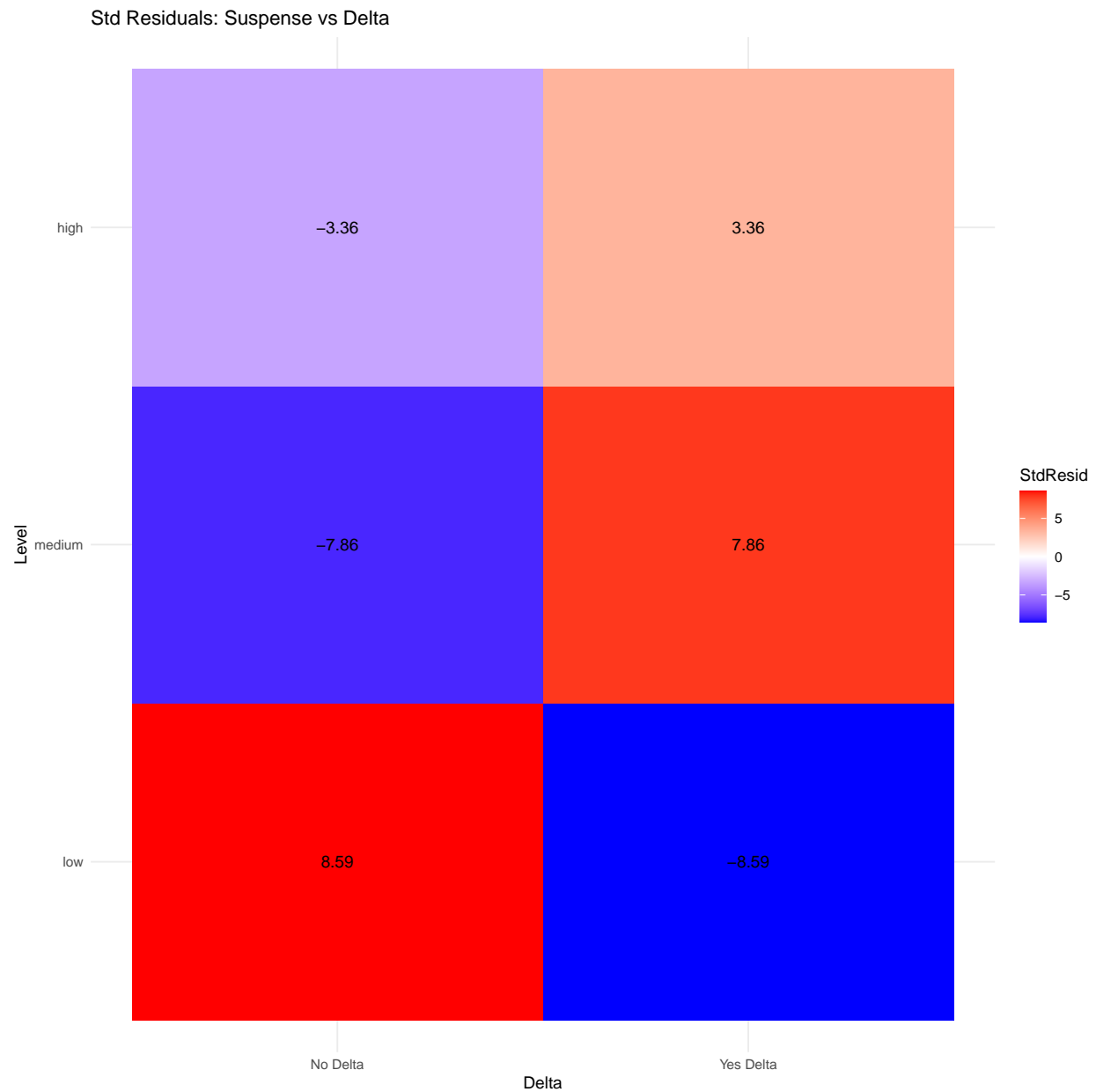
```
df %>%
  group_by(level_suspense, persuasion_success) %>%
  summarise(n = n()) %>%
  group_by(level_suspense) %>%
  mutate(pct = n / sum(n) * 100) %>%
  filter(persuasion_success == 'Yes Delta') %>%
  ggplot(aes(x = level_suspense, y = pct, group=1)) +
  geom_line() + geom_point() +
  labs(title='Yes-Delta % by Suspense Level', y='Percent Yes', x='Suspense Level') +
  theme_minimal()
```

```
## 'summarise()' has grouped output by 'level_suspense'. You can override using
## the '.groups' argument.
```



```
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))
```

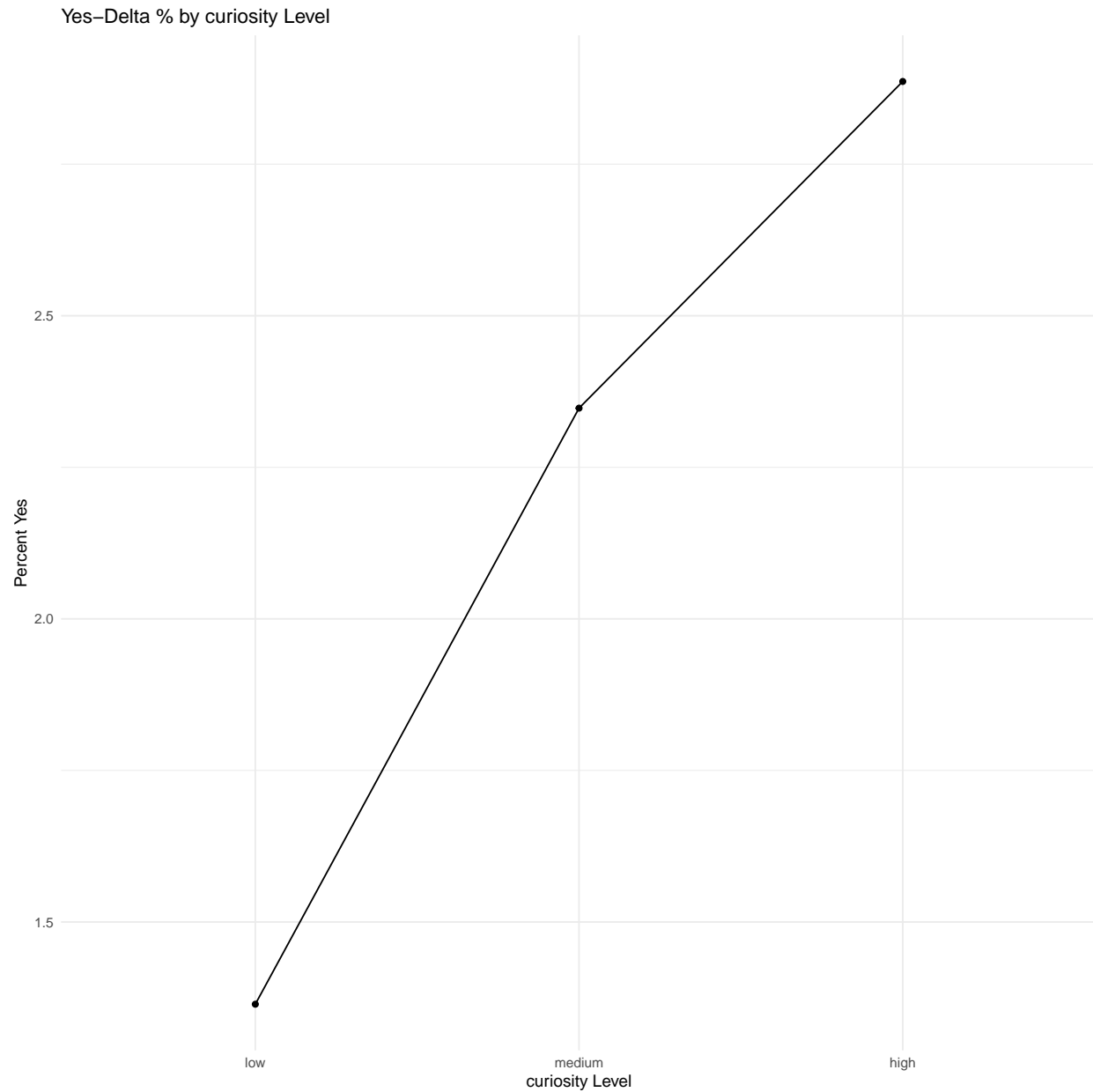
```
## gamma      : 0.285
## std. error  : 0.024
## CI          : 0.238 0.332
```

```
(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
```

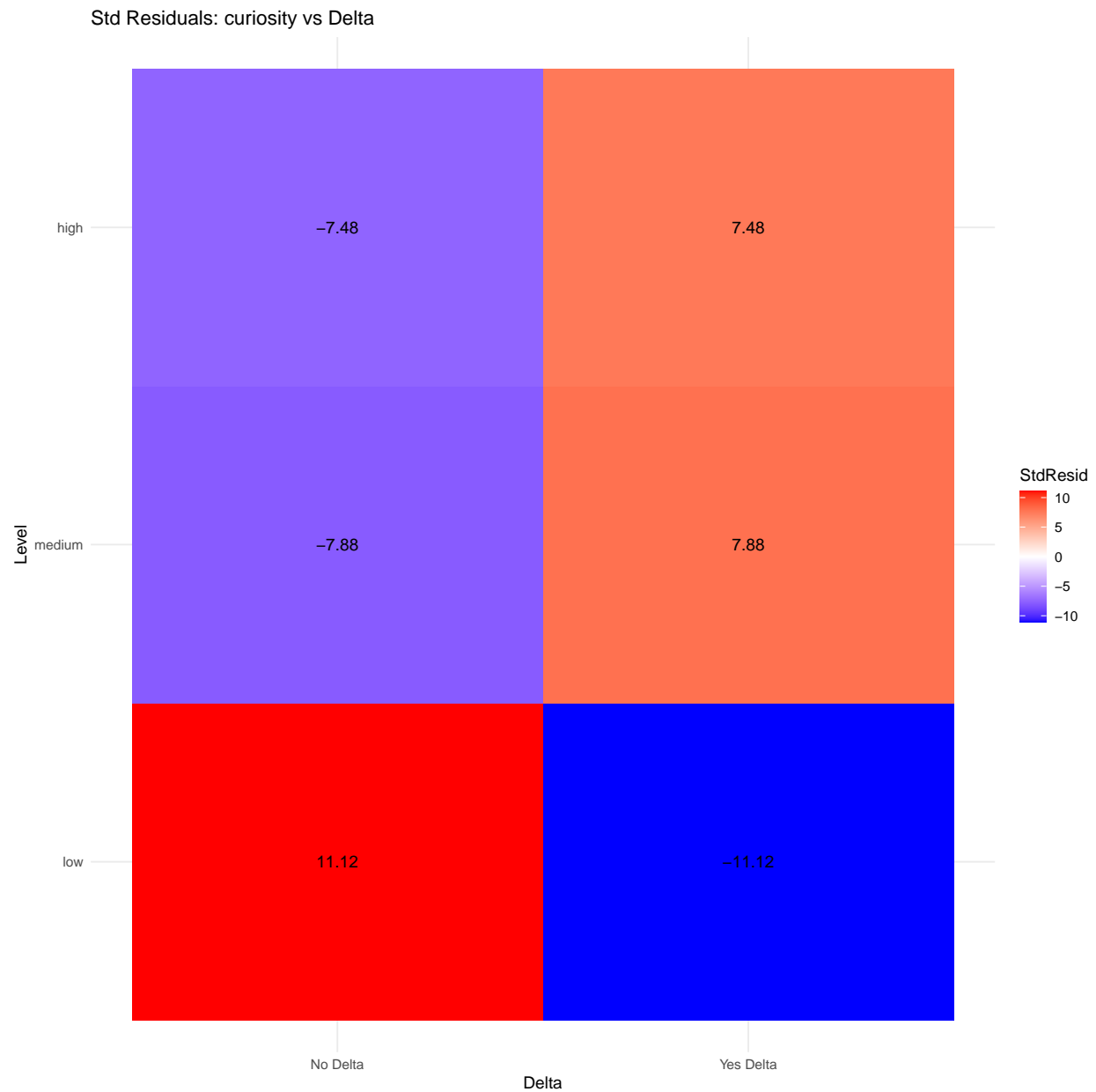
```
df %>%
  group_by(level_curiosity, persuasion_success) %>%
  summarise(n = n()) %>%
  group_by(level_curiosity) %>%
  mutate(pct = n / sum(n) * 100) %>%
  filter(persuasion_success == 'Yes Delta') %>%
  ggplot(aes(x = level_curiosity, y = pct, group=1)) +
  geom_line() + geom_point() +
  labs(title='Yes-Delta % by curiosity Level', y='Percent Yes', x='curiosity Level') +
  theme_minimal()
```

```
## 'summarise()' has grouped output by 'level_curiosity'. You can override using
## the '.groups' argument.
```

```
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))
```

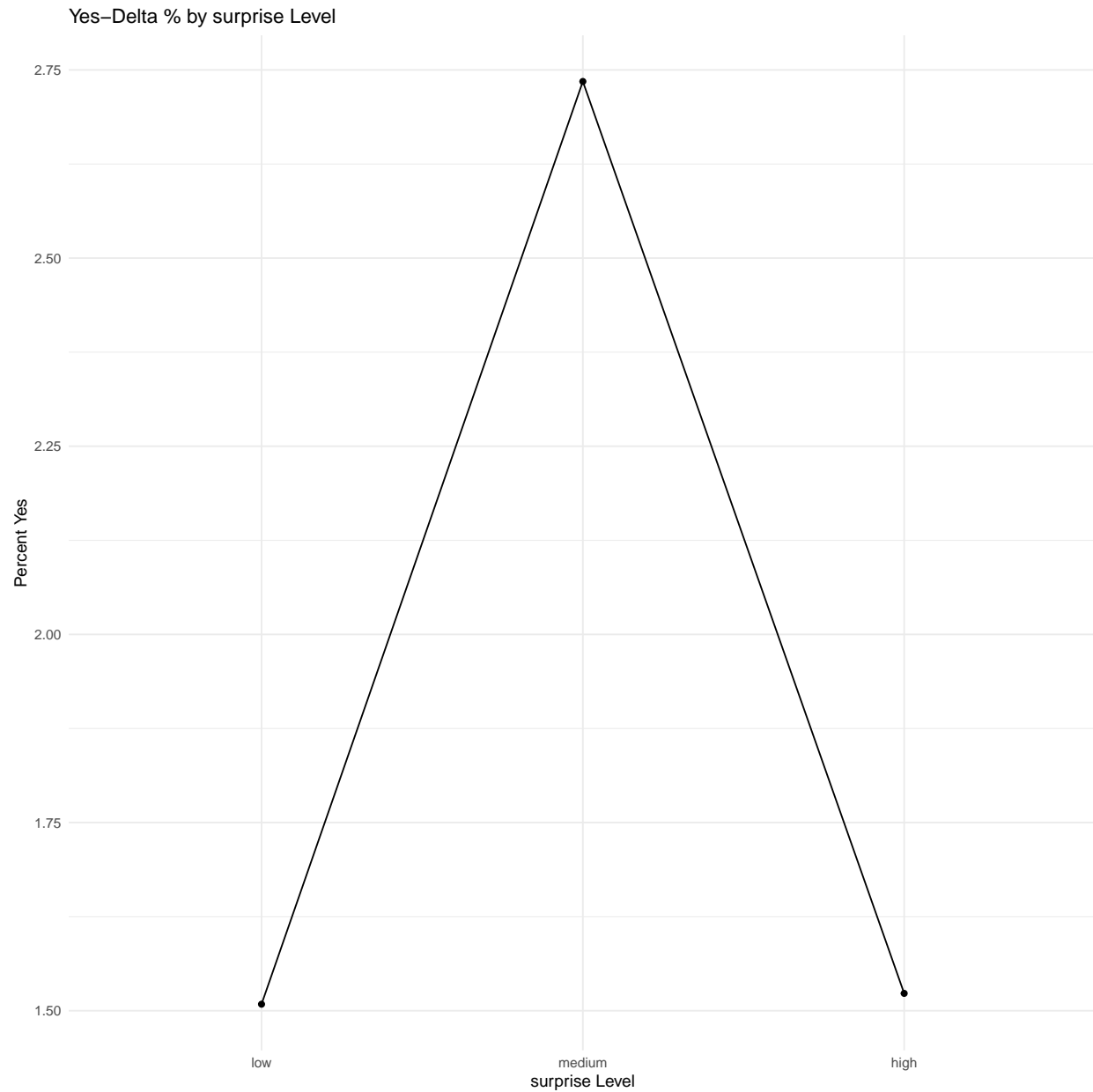
```
## gamma      : 0.207
## std. error  : 0.035
## CI          : 0.139 0.276
```

```
(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
```

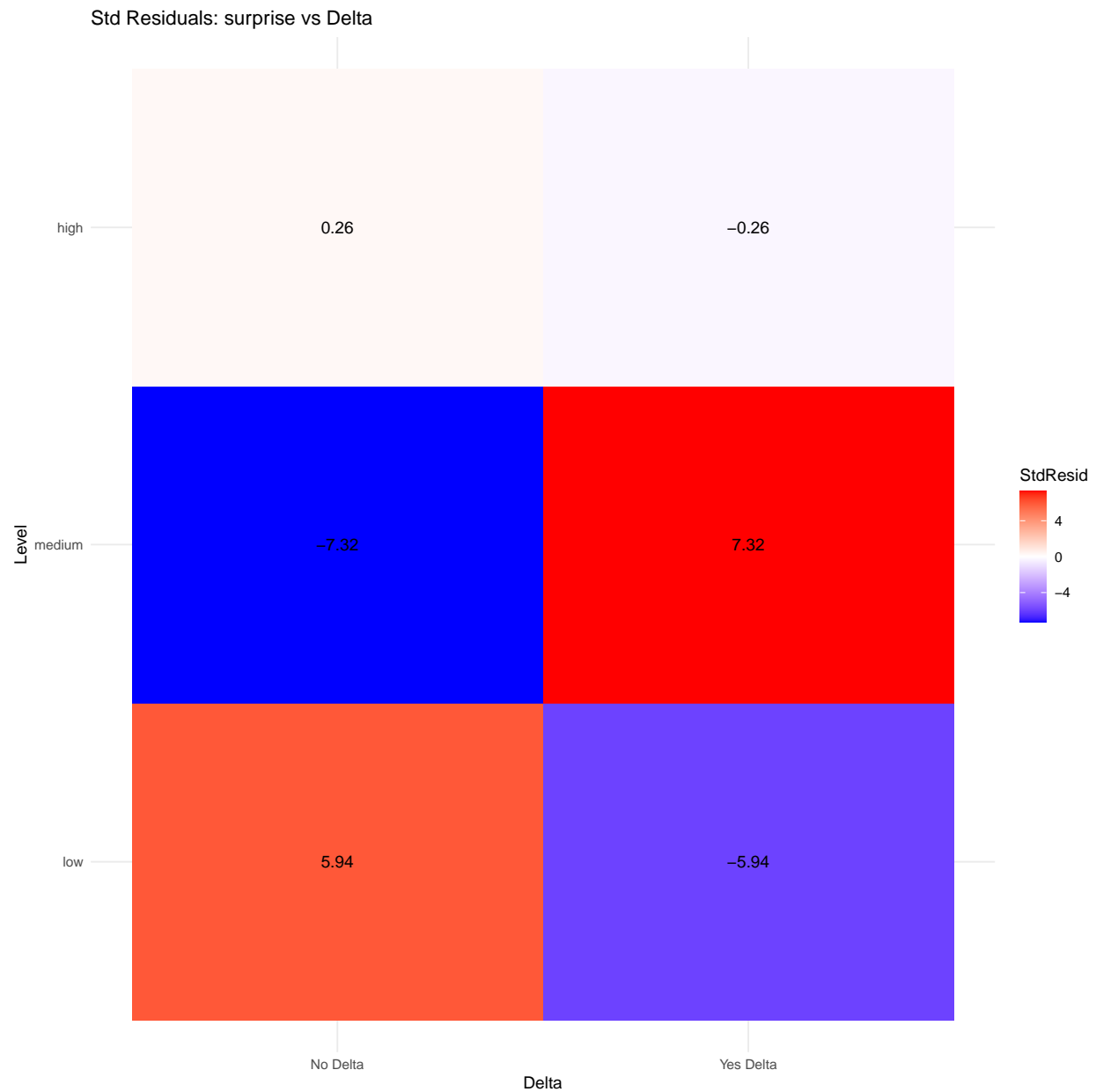
```
df %>%
  group_by(level_surprise, persuasion_success) %>%
  summarise(n = n()) %>%
  group_by(level_surprise) %>%
  mutate(pct = n / sum(n) * 100) %>%
  filter(persuasion_success == 'Yes Delta') %>%
  ggplot(aes(x = level_surprise, y = pct, group=1)) +
  geom_line() + geom_point() +
  labs(title='Yes-Delta % by surprise Level', y='Percent Yes', x='surprise Level') +
  theme_minimal()
```

```
## 'summarise()' has grouped output by 'level_surprise'. You can override using
## the '.groups' argument.
```



```
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()
```

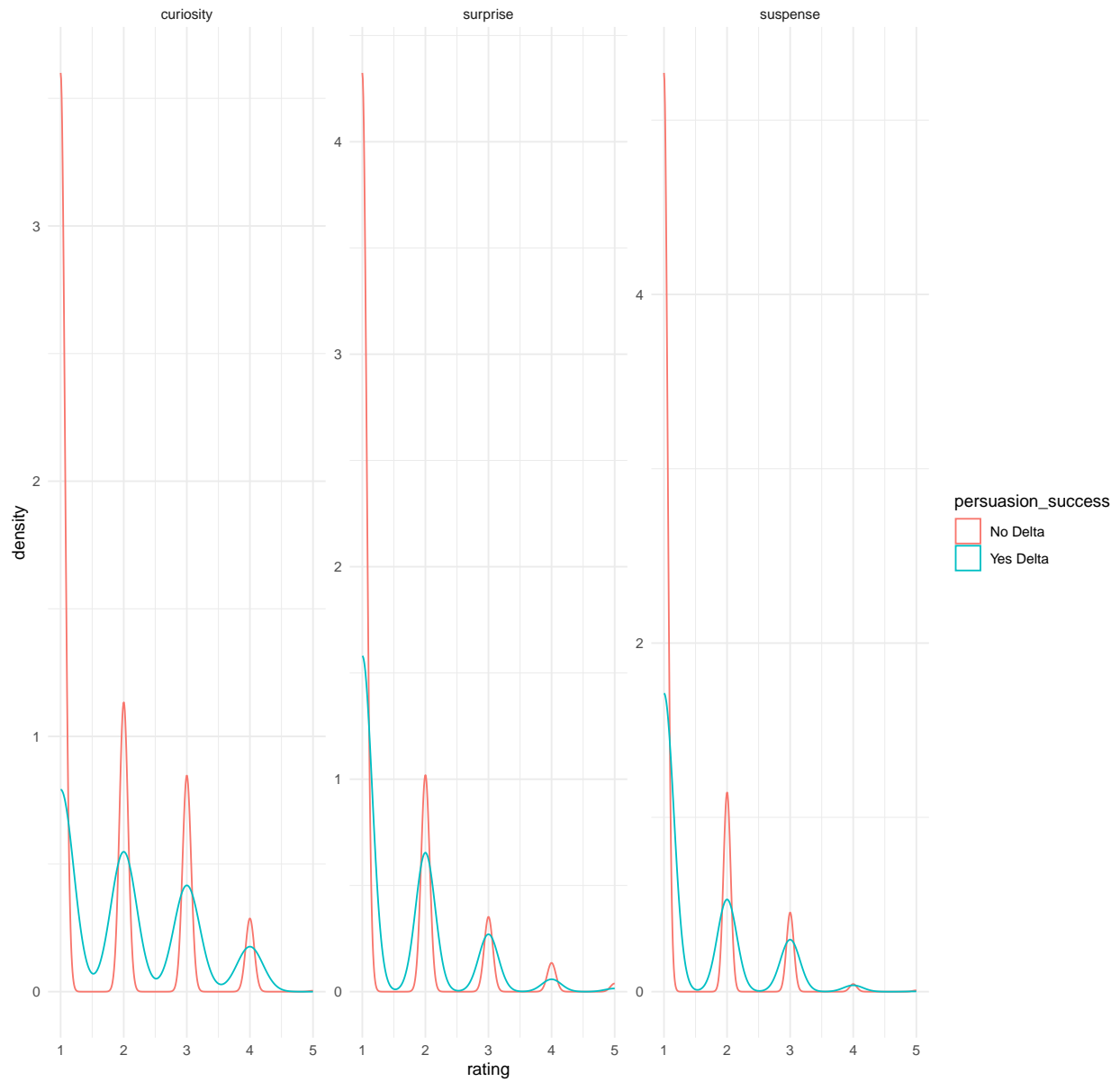


Suspense, Curiosity, Surprise (1-5)

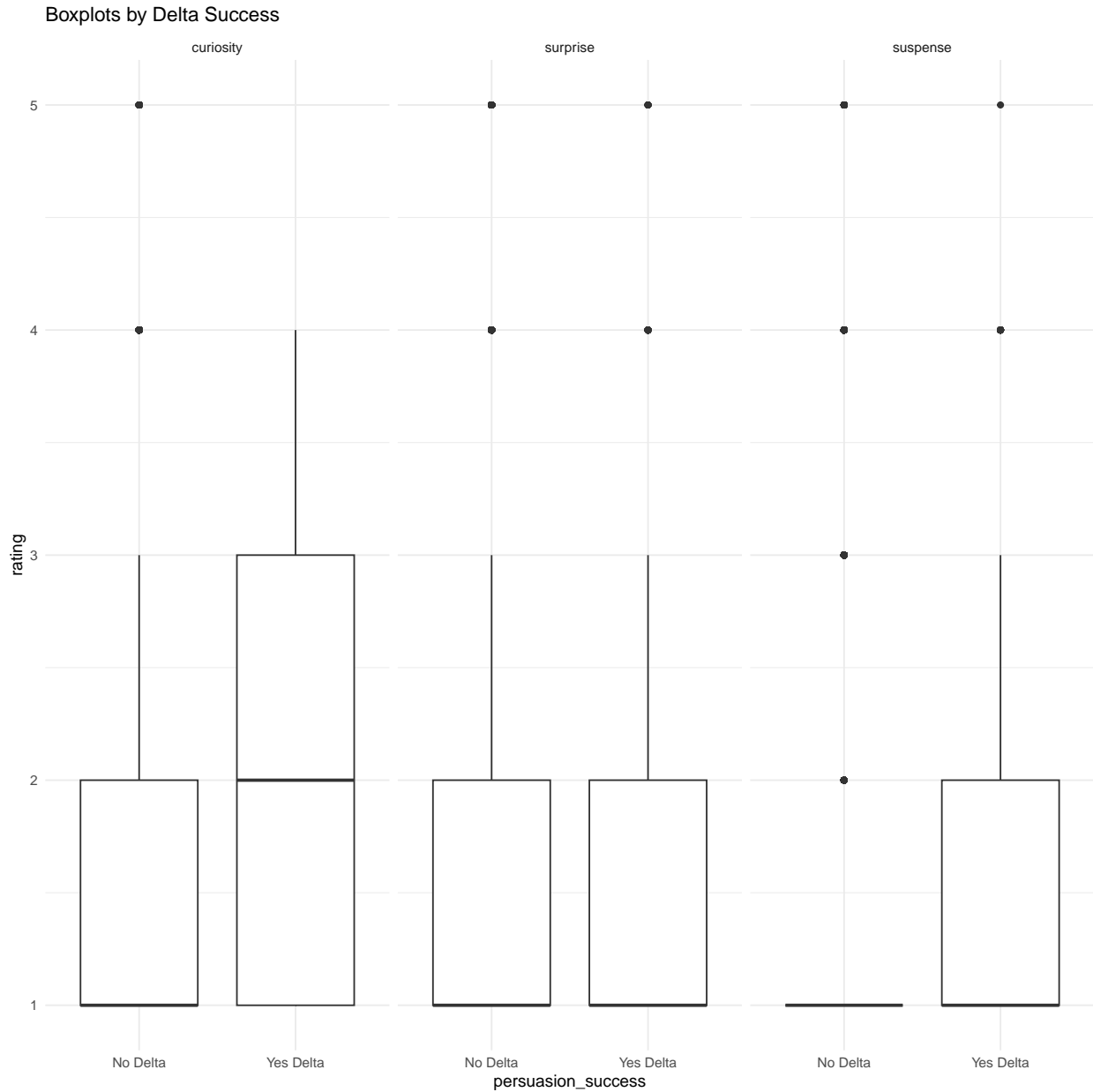
```
df_long <- df %>% select(persuasion_success, suspense, curiosity, surprise) %>%
  pivot_longer(~persuasion_success, names_to='metric', values_to='rating')

ggplot(df_long, aes(x=rating, color=persuasion_success)) +
  geom_density() + facet_wrap(~metric, scales='free') +
  labs(title='Density by Delta Success') + theme_minimal()
```

Density by Delta Success



```
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 +
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
```

```
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_suspensemmedium	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

	OR	CI_low	CI_high
level_surprisehigh	0.182	0.030	0.572
story_classStory:level_suspensemmedium	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_surprisemmedium	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.
```

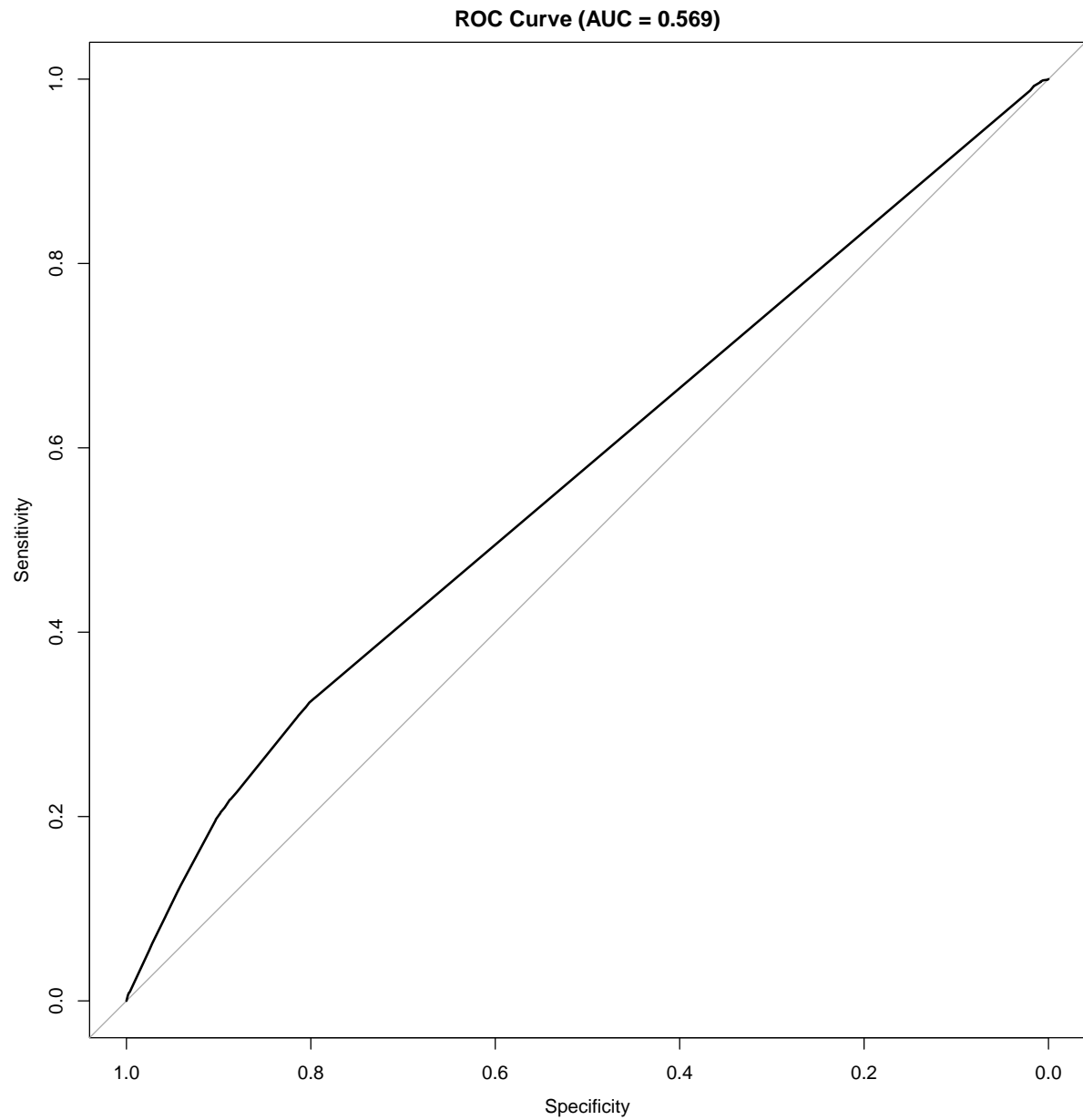
```
##
## Hosmer and Lemeshow goodness of fit (GOF) test
##
## data: as.numeric(df$persuasion_success) - 1, fitted(full_model)
## X-squared = 0.22963, df = 2, p-value = 0.8915
```

```
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))
```

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

```
chisq.test(table_story_class)$expected
```

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

```
chisq.test(table_story_class)
```

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

```
assocstats(table_story_class)$cramer
```

```
## [1] 0.03876783
```

```
#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))
```

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

```
chisq.test(table_level_suspense)$expected
```

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

```
chisq.test(table_level_suspense)
```

```
##
## Pearson's Chi-squared test
##
## data:  table_level_suspense
## X-squared = 74.283, df = 2, p-value < 2.2e-16
```

```
assocstats(table_level_suspense)$cramer
```

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

```
GKgamma(table_level_suspense)
```

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

```
# ordinal trend test (Cochran-Armitage)
# profile plot
# stacked bar chart
# heatmap
```

```
(table_level_curiosity <- table(df$level_curiosity, df$persuasion_success))
```

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

```
chisq.test(table_level_curiosity)$expected
```

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

```
chisq.test(table_level_curiosity)
```

```
##
## Pearson's Chi-squared test
##
## data:  table_level_curiosity
## X-squared = 130.53, df = 2, p-value < 2.2e-16
```

```
assocstats(table_level_curiosity)$cramer
```

```
## [1] 0.03657677
```

```
GKgamma(table_level_curiosity)
```

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

```
# ordinal trend test (Cochran-Armitage)
# profile plot
# stacked bar chart
# heatmap
```

```
(table_level_surprise <- table(df$level_surprise, df$persuasion_success))
```

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

```
chisq.test(table_level_surprise)$expected
```

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

```
chisq.test(table_level_surprise)
```

```
##
## Pearson's Chi-squared test
##
## data:  table_level_surprise
## X-squared = 53.63, df = 2, p-value = 2.262e-12
```

```
assocstats(table_level_surprise)$cramer
```

```
## [1] 0.02344496
```

```
GKgamma(table_level_surprise)
```

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

```
# 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,
  family = binomial, data = df)
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
```

```
#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])
```