Reframing News Through Curation Design

Group 2

2024-03-08

Introduction

Sejin Paik is a graduate student in the Boston University College of Communication Division of Emerging Media Studies. In a pilot study, Reframing the News Through Social Media Curation Design, she aims to explore the potential impacts of social media feed curation on users' perceptions of news items and use the results to help with dissertation. Her research questions were the following: How does the activism feed affect perceptions of news credibility compared to the other surrounding-content newsfeeds? How does the advertisement feed affect perceptions of a) news credibility, b) newsworthiness and c) shareworthiness of the news compared to the other surrounding-content newsfeeds? How does news valence moderate the impact of external content elements on a) news credibility, b) newsworthiness, and c) shareworthiness? How do perceptions of news credibility, newsworthiness and shareworthiness in different curated newsfeeds vary between American and Chinese participants? From those questions, she formulated two main hypotheses. Memes will produce a greater level of a) credibility, b) newsworthiness, and c) shareworthiness perceptions of the news compared to the news-only newsfeed. Activism will produce a greater level of a) newsworthiness and b) shareworthiness perception of the news compared to the other content surrounding newsfeeds.

Sejin's data came from two different samples. The first sample was composed of 661 participants from the United States and the second sample was composed of 484 participants from China. Participants in both samples were randomly assigned to one of eight conditions according to a 4 by 2 factorial design. There were two independent variables, one with four levels and a second with two levels. The first independent variable was feed content type, which was one of the following: activism, advertisements, memes, and news. The second independent variable was story valence, which was either positive or negative. She also provided a copy of the manuscript for her paper, which included some of her background research and hypotheses, a copy of the survey that was distributed to the participants, and the SPSS output for her analyses.

Each newsfeed condition consists of seven posts: a positive or negative news story positioned fourth in the vertical newsfeed stack, surrounded by one of four content elements (memes, ads, activism, or additional news stories). The feeds were presented with topic filtering to reduce potential confounding variables and the topic is about artificial intelligence (AI), a relatively politically neutral subject.

The client requested assistance with a few tasks. First, she requested a duplication of ANOVA and ANCOVA tests that were performed to test her hypotheses to ensure that accurate results were obtained. Second, she requested that the results of the assumption checking be verified, as well as assistance with potentially modifying the analysis depending on whether the appropriate assumptions were met. Third, she asked that if there any potentially alternative or additional methods of analysis that might be more appropriate to answer her hypotheses and research questions, that those be recommended to her.

```
## Warning: package 'readr' was built under R version 4.3.2

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode

## Warning: package 'ggplot2' was built under R version 4.3.3

##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine

## Warning: package 'effects' was built under R version 4.3.3

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

##
## Attaching package: 'MASS'

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

## Warning: package 'kableExtra' was built under R version 4.3.2

##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows

## Rows: 484 Columns: 351
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (8): StartDate, EndDate, IPAddress, RecordedDate, ResponseId, Distribu...
## dbl (324): Status, Progress, Durationinseconds, Finished, LocationLatitude, ...
## lgl  (19): RecipientLastName, RecipientFirstName, RecipientEmail, ExternalRe...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
## Rows: 673 Columns: 403
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (14): Date, EndDate, IPAddress, RecordedDate, ResponseId, DistributionC...
## dbl (367): Status, Progress, Durationinseconds, Finished, LocationLatitude, ...
## lgl  (22): RecipientLastName, RecipientFirstName, RecipientEmail, ExternalRe...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Data preparation

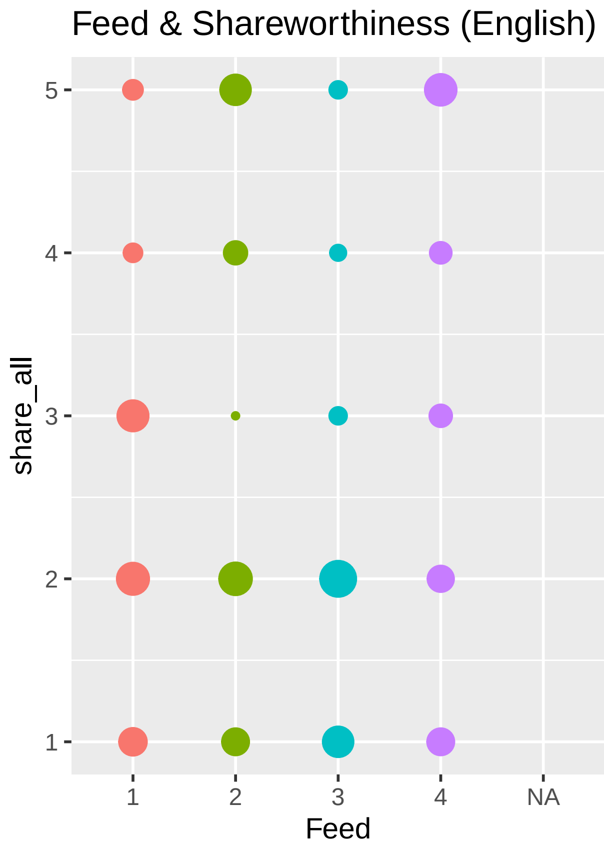
The surveys were administered to participants via Qualtrics and delivered to the consulting group as two separate .csv files, one for the American sample and one for the Chinese sample. They both contained variables such as the feed content and valence conditions the participants were assigned to, ratings of news credibility, ratings of newsworthiness, and ratings of news shareworthiness.

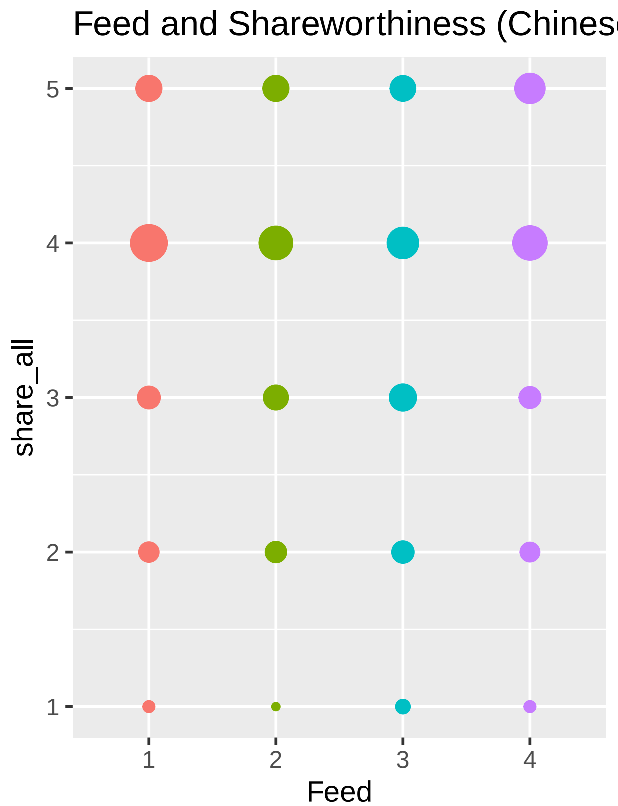
One of the main steps required as part of the data preparation was standardizing the labels for the main variables of interest in the data, since there were some labeling discrepancies between the two datasets. A second step involved changing certain variables to the appropriate data classes. For example, the "valence" variable had to be changed from numeric to factor. The next step was to filter the relevant variables for the analysis from each of the datasets to include in the analysis; both sets included hundreds of columns, many of which were unrelated to the relevant objectives. Please note that for our analysis, we are performing tests on both datasets separately.

```
##             credavg_all nwavg_all share_all
## credavg_all   1.0000000 0.6336139 0.6220074
## nwavg_all     0.6336139 1.0000000 0.5359845
## share_all     0.6220074 0.5359845 1.0000000
```

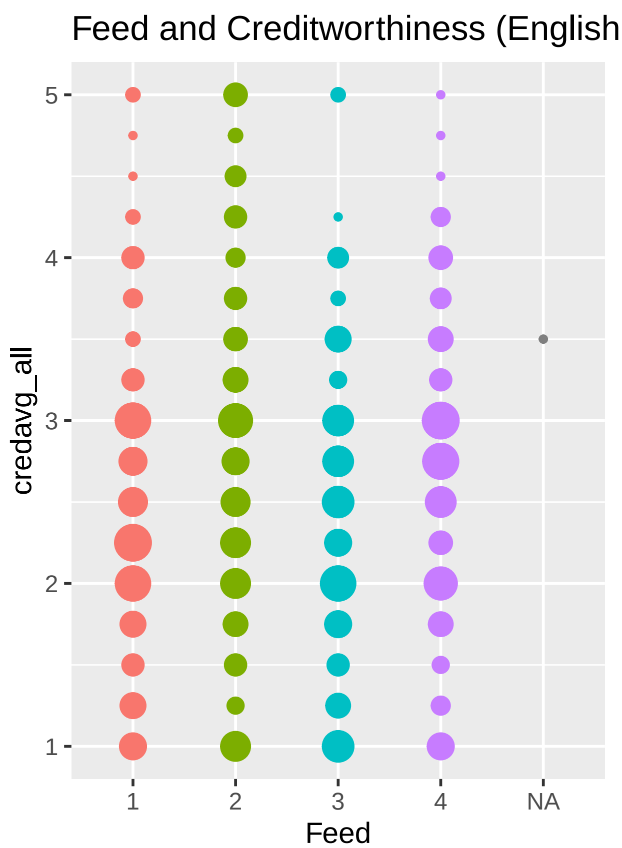
```
##             credavg_all nwavg_all share_all
## credavg_all   1.0000000 0.6500306 0.5593801
## nwavg_all     0.6500306 1.0000000 0.4560854
## share_all     0.5593801 0.4560854 1.0000000
```

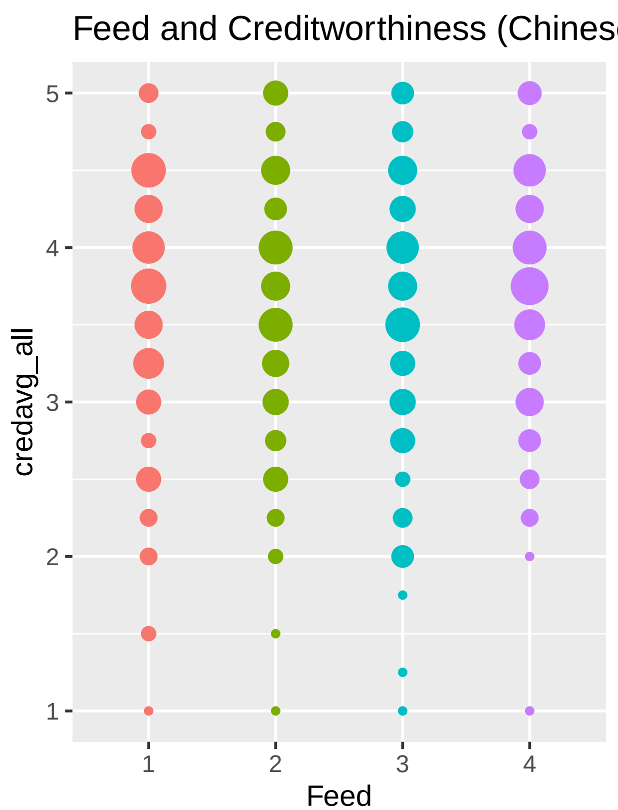
There are positive moderate correlations among the three variables. This suggests that our statistical analysis for each outcome variables (credibiitity, newsworthiness, and shareworthiness) might produce very similar results. Also, to simplify the model, we can also consider combining the three measurements into one score, which can be a new column in the dataset. # Statistical Analysis: ## ANOVA Assumption Check:



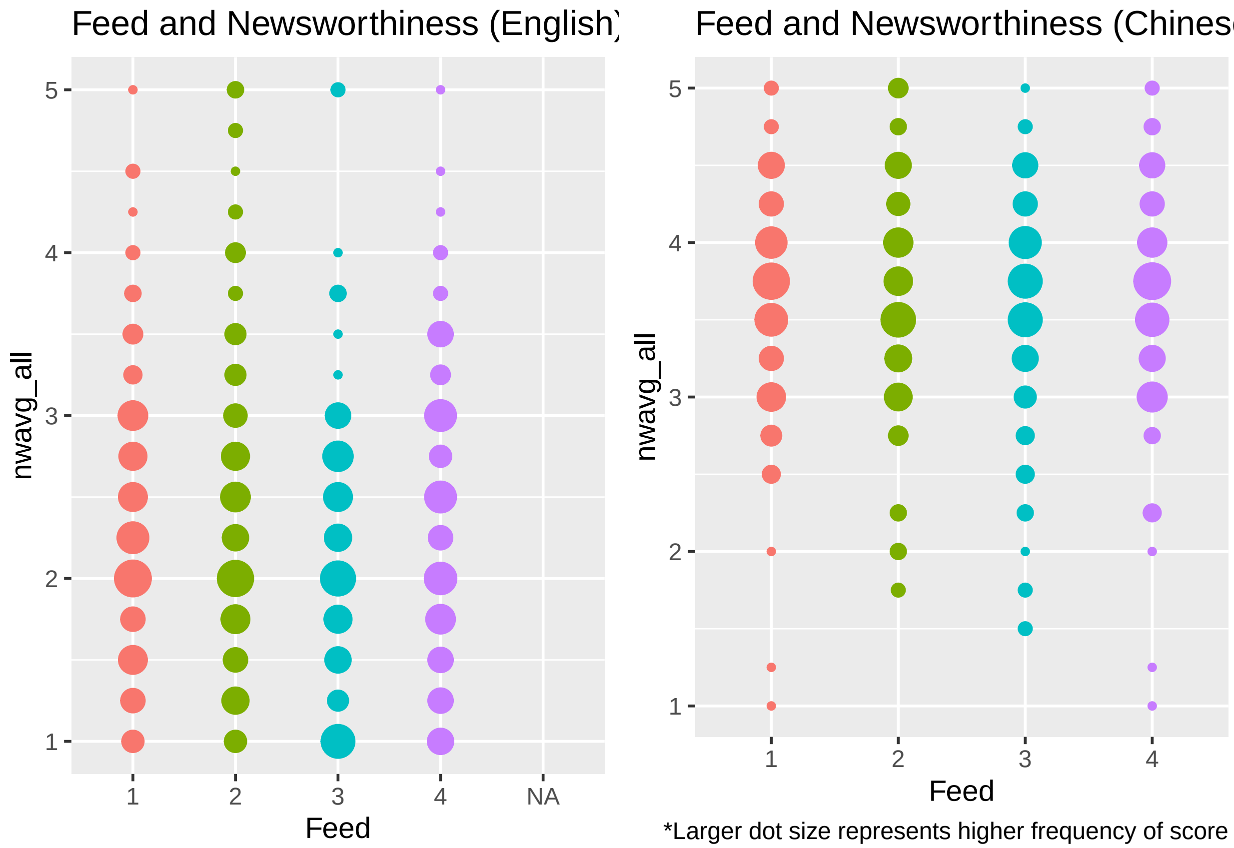


*Larger dot size represents higher frequency of score





*Larger dot size represents higher frequency of score



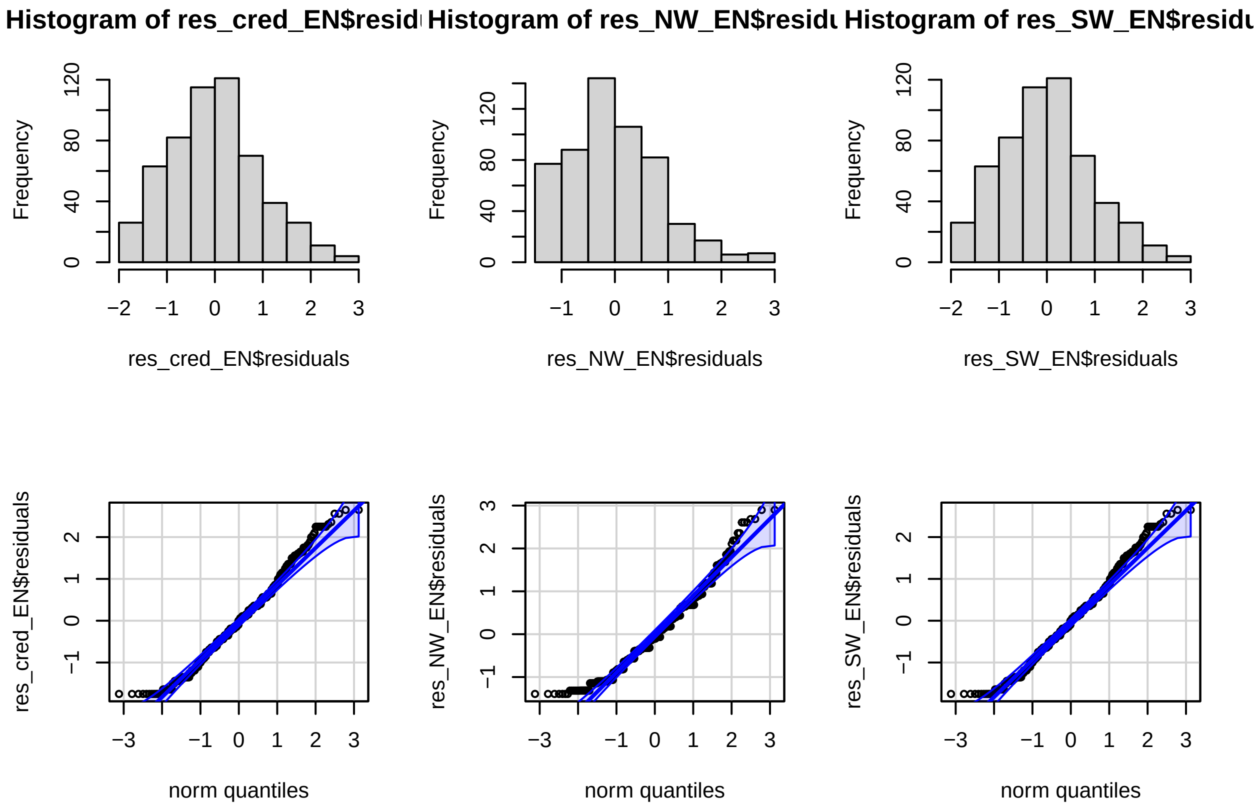
Hypothesis 2

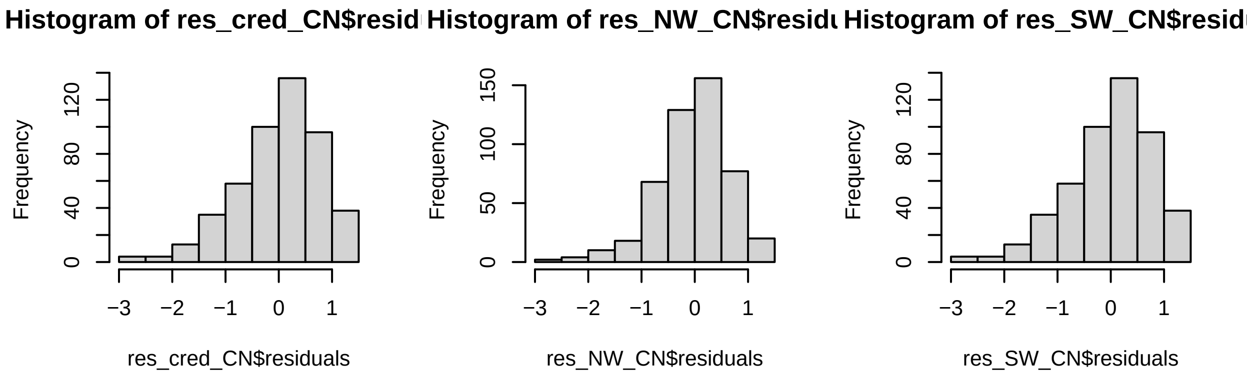
Activism will produce a greater level of: a) Newsworthiness b) Share-worthiness Compared with the other 3 feeds.

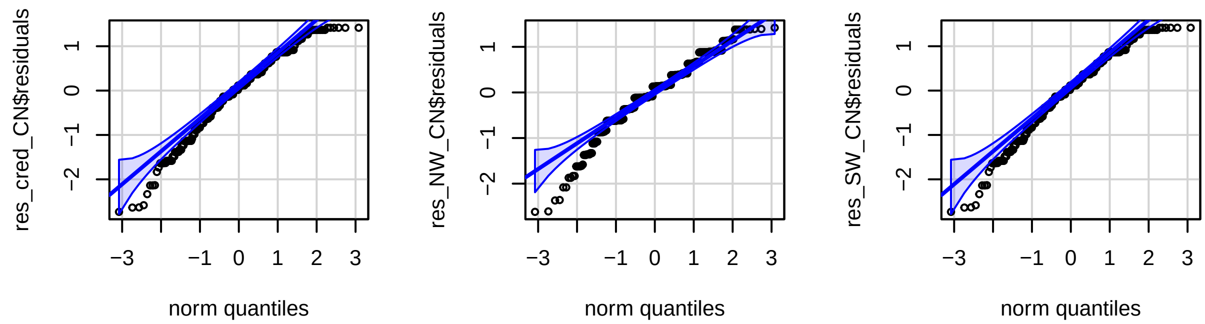
This hypothesis was originally tested with ANCOVAs, with the feed content types and valences treated as independent variables and the following items as covariates: participants' frequency of sharing news posts on social media, attitudes on technology's impact (positive or negative) on society, and feelings toward AI automation. However, the client requested that the assumptions for these tests be repeated in order to make sure that these tests were appropriate.

Hypothesis 2 Assumption checks

The first assumption to be checked was the normality of residuals.This assumption was for the Chinese and English datasets individually, since the client ran the tests on each sample separately; these assumption checks were repeated. The assumption was also checked for the combined dataset, since it is appropriate to check given that the data for the two groups were capable of being combined and the pertinent hypothesis does not explicitly reference a distinction between the two groups. The assumption was also checked for the Credibility measure, since this is required for some of the tests performed later in the analysis.First, visuals were created:


Histogram of res_cred_EN$resid
Histogram of res_NW_EN$resid
Histogram of res_SW_EN$resid
Frequency
res_cred_EN$residuals
res_NW_EN$residuals
res_SW_EN$residuals
norm quantiles





As is shown in the plots above,the residuals look skewed in opposite directions for the English and Chinese datasets for all three outcomes of interest. Next, normality of the dependent variables was formally tested with the Shapiro-Wilk Test.

```
##
##  Shapiro-Wilk normality test
##
## data:  res_cred_CN$residuals
## W = 0.97264, p-value = 7.286e-08

##
##  Shapiro-Wilk normality test
##
## data:  res_NW_CN$residuals
## W = 0.96047, p-value = 4.101e-10

##
##  Shapiro-Wilk normality test
##
## data:  res_SW_CN$residuals
## W = 0.97264, p-value = 7.286e-08

##
##  Shapiro-Wilk normality test
##
## data:  res_cred_EN$residuals
## W = 0.98423, p-value = 9.945e-06

##
```

```
##  Shapiro-Wilk normality test
##
## data:  res_NW_EN$residuals
## W = 0.96324, p-value = 1.395e-10
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_SW_EN$residuals
## W = 0.98423, p-value = 9.945e-06
```

Another assumption that had to be checked was the homogeneity of variance. Levene's Test was performed to verify that this assumption holds true across the four feed conditions and the two valence conditions; it was applied for both the English and Chinese datasets, to replicate what the client did.

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   3  1.4586 0.2249
##       553
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   3  0.5177 0.6703
##       480
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value   Pr(>F)
## group   3  4.8969 0.002285 **
##       553
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   3  1.8656 0.1345
##       480
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value  Pr(>F)
## group   3  3.6119 0.01321 *
##       553
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   3  1.3355 0.2621
##       480
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value   Pr(>F)
## group   1  36.552 2.73e-09 ***
##       555
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  0.8978 0.3438
```

```
##        482

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  0.2865 0.5927
##       555

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1   8e-04 0.9781
##       482

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value    Pr(>F)
## group   1  14.044 0.0001973 ***
##       556
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   1  1.2809 0.2583
##       482
```

The Levene's Tests yielded mixed results. The test fails for a majority of the variables with the English dataset, with the exception of the newsworthiness across content feed types and shareworthiness between the two valence types. The tests passed across all variables for the Chinese dataset. The tests also passed across all variables for the combined dataset, with the exception of the assumption of homogeneity of variance for newsworthiness between the two valence conditions.

Analysis:

The assumptions for the tests utilized were not met in all cases. Assumptions of normality for the outcomes of interest were not met. When the tests were replicated for each of the two samples separately, they failed for all outcome variables of interest;they also failed across all variables when the data was aggregated. Assumptions of homogeneity were not met for several of the variables of interest. The analysis replicated the findings of the client in terms of the English dataset. This means that alternative tests are required for some of the hypothesis testing and to answer the research questions appropriately.

However, since the outcome variables are ordinal variables, applying ANOVA or other non-parametric, non-normal tests might not be appropriate since those tests treat the outcome variables as continuous values. Even though in practice, there are many academic disciplines that use ANOVA or statistical tests that treat the outcome variables to be continuous even though they are ordinal, but from the statistical perspective, the method is not correct. Hence, it is recommended to perform statistical tests that are designed for ordinal outcomes. For this reason, we consider using ordinal logistic regression. Using this method might not necessarily answer specific research questions that the client presents but our results can be used as a first step to explore the dataset and see the relationships among the variables in a quantitative way besides visualization.

Modeling with Proportional-Odds Logistic Regression:

As suggested in the previous section, we attempt to fit the proportional-odds logistic regression on each outcome variables (credibility, newsworthiness, and shareworthiness). The independent variables are surrounding content-type and feed valance. Note that in the dataset, the client has converted feed types into numerical values.

1 - news-mixed feeds 2 - meme-mixed feeds 3 - activism-mixed feeds 4 - ad-mixed feeds

and feed valence to be 1- positive news 2- negative news

Interpretation

The following table lays out the coefficients of the odds ratios and the 95% confidence intervals of the coefficients for each of the three dependent variables of interest for both the English and Chinese datasets:

Odds Ratios for Credibility (English)

	OR	2.5%	97.5%
Feed 2	1.63	1.05	2.53
Feed 3	0.85	0.55	1.31
Feed 4	1.54	1.00	2.38
Valence 2	1.23	0.91	1.68

Odds Ratios for Newsworthiness (English)

	OR	2.5%	97.5%
Feed 2	1.09	0.71	1.68
Feed 3	0.66	0.43	1.02
Feed 4	1.15	0.75	1.78
Valence 2	0.51	0.37	0.70

Odds Ratios for Shareworthiness (English)

	OR	2.5%	97.5%
Feed 2	1.29	0.83	2.01
Feed 3	0.78	0.50	1.21
Feed 4	1.15	0.97	2.35
Valence 2	0.85	0.62	1.17

Odds Ratios for Credibility (Chinese)

	OR	2.5%	97.5%
Feed 2	0.98	0.62	1.55
Feed 3	0.95	0.60	1.51
Feed 4	1.24	0.80	1.94
Valence 2	0.71	0.51	0.98

Odds Ratios for Newsworthiness (Chinese)

	OR	2.5%	97.5%
Feed 2	0.97	0.61	1.55
Feed 3	1.07	0.68	1.69
Feed 4	1.01	0.65	1.58
Valence 2	1.43	1.03	1.99

Odds Ratios for Shareworthiness (Chinese)

	OR	2.5%	97.5%
Feed 2	0.88	0.55	1.42
Feed 3	0.76	0.47	1.23
Feed 4	1.46	0.91	2.34
Valence 2	1.11	0.79	1.56

Interpretation for English Dataset

Based on the output provided for the Proportional Odds Logistic Regression model, here's the interpretation:

Credibility Variable

1. Feed Variable: For feed variable, only
   - For "Feed" category 2 (memes), the coefficient is 0.488 with a p-value of approximately 0.029. This indicates that, compared to the reference category (news), being exposed to memes is associated with a statistically significant increase in the log odds of being in a higher category of credavg_all. The odds ratio associated with memes (as.factor(Feed)2) is approximately 1.629. This suggests that the odds of being in a higher category of credibility are approximately 1.629 times higher for individuals exposed to memes compared to those exposed to news.
   - For "Feed" category 3 (act), the coefficient is -0.165 with a p-value of approximately 0.453. This suggests that there is no statistically significant difference in the log odds of being in a higher category of credavg_all between individuals exposed to act content and those exposed to news content.
   - For "Feed" category 4 (adv), the coefficient is 0.432 with a p-value of approximately 0.0499. This indicates that, compared to the reference category (news), exposure to advertisements is associated with a statistically significant increase in the log odds of being in a higher category of credavg_all.The odds ratio associated with advertisements (as.factor(Feed)4) is approximately 1.540. This suggests that the odds of being in a higher category of credavg_all are approximately 1.555 times higher for individuals exposed to advertisements compared to those exposed to news.
2. Valence Variable:
   - For "Valence" category 2 (negative news), the coefficient is 0.209 with a p-value of approximately 0.186. This suggests that, the effect of valence variable for credibility is not statistically significant.

Newsworthiness Variable

1. Feed Variable:
   - For "Feed" category 2 (memes), the coefficient is 0.087 with a p-value of approximately 0.692. This suggests that there is no statistically significant difference in the log odds of perceiving content as newsworthy between individuals exposed to memes and those exposed to news (reference category).
   - For "Feed" category 3 (act), the coefficient is -0.412 with a p-value of approximately 0.063. This suggests that there is no statistically significant difference in the log odds of being in a higher

category of newsworthiness between individuals exposed to act content and those exposed to news content.
    - For "Feed" category 4 (adv), the coefficient is 0.144 with a p-value of approximately 0.516. This indicates that there is no statistically significant difference in the log odds of perceiving content as newsworthy between individuals exposed to advertisements and those exposed to news (reference category).
2. Valence Variable:
    - For "Valence" category 2 (negative news), the coefficient is -0.681 with a p-value of approximately 0.0000271. This indicates that, compared to positive news (reference category), exposure to negative news is associated with a statistically significant decrease in the log odds of perceiving content as newsworthy. The odds ratio associated with negative news (as.factor(valence)2) is approximately 0.506. This suggests that the odds of perceiving content as newsworthy are approximately 0.506 times lower for individuals exposed to negative news compared to those exposed to positive news.

Shareworthiness Variable

1. **Feed Variable:**
    - For "Feed" category 2 (memes), the coefficient is 0.258 with a standard error of approximately 0.224. The p-value is approximately 0.249. This suggests that there is no statistically significant difference in the log odds of perceiving content as shareworthy between individuals exposed to memes and those exposed to news (reference category).
    - For "Feed" category 3 (act), the coefficient is -0.249 with a standard error of approximately 0.225. The p-value is approximately 0.267. This indicates that there is no statistically significant difference in the log odds of perceiving content as shareworthy between individuals exposed to act content and those exposed to news (reference category).
    - For "Feed" category 4 (adv), the coefficient is 0.410 with a standard error of approximately 0.226. The p-value is approximately 0.070. This suggests that there is no statistically significant difference in the log odds of perceiving content as shareworthy between individuals exposed to advertisements and those exposed to news (reference category).
2. **Valence Variable:**
    - For "Valence" category 2 (negative news), the coefficient is -0.161 with a standard error of approximately 0.161. The p-value is approximately 0.315. This suggests that there is no statistically significant difference in the log odds of perceiving content as shareworthy between individuals exposed to negative news and those exposed to positive news (reference category).

Interpretation for Chinese Dataset

Based on the output provided for the Proportional Odds Logistic Regression model, here's the interpretation:

Credibility Variable

1. Feed Variable:
    - For "Feed" category 2, the coefficient is -0.019 with a standard error of approximately 0.233. The p-value is approximately 0.932. This suggests that there is no statistically significant difference in the log odds of perceived credibility between individuals exposed to content categorized as "Feed" category 2 (memes) and those exposed to the reference category (presumably news).
    - For "Feed" category 3, the coefficient is -0.053 with a standard error of approximately 0.236. The p-value is approximately 0.822. This indicates that there is no statistically significant difference in the log odds of perceived credibility between individuals exposed to content categorized as "Feed" category 3 (act) and those exposed to the reference category.
    - For "Feed" category 4, the coefficient is 0.219 with a standard error of approximately 0.226. The p-value is approximately 0.332. This suggests that there is no statistically significant difference in the log odds of perceived credibility between individuals exposed to content categorized as "Feed" category 4 (adv) and those exposed to the reference category.
2. Valence Variable:
    - For "Valence" category 2 (negative news), the coefficient is -0.349 with a standard error of

approximately 0.166. The p-value is approximately 0.033. The odds ratio associated with negative valence (as.factor(valence)2) is approximately 1.244. This suggests that the odds of perceiving content as credible are approximately 1.244 times lower for individuals exposed to negative valence content compared to those exposed to positive valence content.

Newsworthiness Variable

Thank you for providing the output for the Proportional Odds Logistic Regression model with the dependent variable "nwavg_all" and the independent variables "Feed" and "valence". Let's interpret the results:

1. Feed Variable:

   - For "Feed" category 2, the coefficient is approximately -0.028 with a p-value of approximately 0.906. This suggests that there is no statistically significant difference in the log odds of perceived newsworthiness between individuals exposed to memes and those exposed to a reference category (presumably news).
   - For "Feed" category 3, the coefficient is approximately 0.072 with a p-value of approximately 0.757. This indicates that there is no statistically significant difference in the log odds of perceived newsworthiness between individuals exposed to act content and those exposed to the reference category.
   - For "Feed" category 4, the coefficient is approximately 0.011 with a p-value of approximately 0.961. This suggests that there is no statistically significant difference in the log odds of perceived newsworthiness between individuals exposed to advertisements and those exposed to the reference category.

2. Valence Variable:

   - For "Valence" category 2 (negative valence), the coefficient is approximately 0.357 with a p-value of approximately 0.033. This indicates that there is a statistically significant difference in the log odds of perceived newsworthiness between individuals exposed to negative valence content and those exposed to positive valence content (reference category). The odds ratio associated with negative valence (as.factor(valence)2) is approximately 1.429. This suggests that the odds of perceiving content as newsworthy are approximately 1.429 times higher for individuals exposed to negative valence content compared to those exposed to positive valence content.

Shareworthiness Variable

Thank you for providing the output for the Proportional Odds Logistic Regression model with the dependent variable "share_all" and the independent variables "Feed" and "valence". Let's interpret the results:

1. Feed Variable:
   - For "Feed" category 2, the coefficient is approximately -0.126 with a p-value of approximately 0.604. This suggests that there is no statistically significant difference in the log odds of perceived shareworthiness between individuals exposed to memes and those exposed to a reference category (presumably news).
   - For "Feed" category 3, the coefficient is approximately -0.272 with a p-value of approximately 0.268. This indicates that there is no statistically significant difference in the log odds of perceived shareworthiness between individuals exposed to act content and those exposed to the reference category.
   - For "Feed" category 4, the coefficient is approximately 0.375 with a p-value of approximately 0.119. This suggests that there is no statistically significant difference in the log odds of perceived shareworthiness between individuals exposed to advertisements and those exposed to the reference category.
2. Valence Variable:
   - For "Valence" category 2 (negative valence), the coefficient is approximately 0.101 with a p-value of approximately 0.561. This indicates that there is no statistically significant difference in the log

odds of perceived shareworthiness between individuals exposed to negative valence content and those exposed to positive valence content (reference category).

Appendix

English data:

```
Data_EN <- na.omit(Data_EN)
Data_EN <- Data_EN %>%
  mutate(across(c("credavg_all", "Feed", "valence"), factor))
model1_cred_EN <- polr(as.factor(credavg_all) ~ as.factor(Feed) + as.factor(valence), data = Data_EN)
model1_news_EN <- polr(as.factor(nwavg_all) ~ as.factor(Feed) + as.factor(valence), data = Data_EN)
model1_share_EN <- polr(as.factor(share_all) ~ as.factor(Feed) + as.factor(valence), data = Data_EN)
summary(model1_cred_EN)
```

```
##
## Re-fitting to get Hessian
```

```
## Call:
## polr(formula = as.factor(credavg_all) ~ as.factor(Feed) + as.factor(valence),
##     data = Data_EN)
##
## Coefficients:
##                        Value Std. Error t value
## as.factor(Feed)2     0.4882     0.2237  2.1820
## as.factor(Feed)3    -0.1658     0.2209 -0.7507
## as.factor(Feed)4     0.4321     0.2204  1.9601
## as.factor(valence)2  0.2088     0.1579  1.3224
##
## Intercepts:
##           Value   Std. Error t value
## 1|1.25    -2.0622  0.2150    -9.5940
## 1.25|1.5 -1.6396  0.1981    -8.2777
## 1.5|1.75 -1.3174  0.1889    -6.9730
## 1.75|2    -0.8761  0.1800    -4.8681
## 2|2.25    -0.2659  0.1736    -1.5317
## 2.25|2.5  0.1115  0.1730     0.6447
## 2.5|2.75  0.4960  0.1748     2.8379
## 2.75|3    0.8786  0.1781     4.9331
## 3|3.25    1.5418  0.1884     8.1827
## 3.25|3.5  1.7970  0.1945     9.2410
## 3.5|3.75  2.1496  0.2048    10.4986
## 3.75|4    2.4815  0.2169    11.4411
## 4|4.25    3.0269  0.2450    12.3548
## 4.25|4.5  3.4946  0.2793    12.5104
## 4.5|4.75  3.8660  0.3156    12.2484
## 4.75|5    4.2060  0.3582    11.7426
##
## Residual Deviance: 2564.50
## AIC: 2604.50
```

```
summary(model1_news_EN)
```

```
##
## Re-fitting to get Hessian
```

```
## Call:
## polr(formula = as.factor(nwavg_all) ~ as.factor(Feed) + as.factor(valence),
##     data = Data_EN)
##
## Coefficients:
##                         Value Std. Error t value
## as.factor(Feed)2     0.08743     0.2209  0.3958
## as.factor(Feed)3    -0.41239     0.2216 -1.8609
## as.factor(Feed)4     0.14353     0.2212  0.6488
## as.factor(valence)2 -0.68069     0.1622 -4.1968
##
## Intercepts:
##          Value    Std. Error t value
## 1|1.25    -2.7994   0.2291   -12.2168
## 1.25|1.5  -2.2013   0.2045   -10.7618
## 1.5|1.75  -1.7023   0.1908    -8.9238
## 1.75|2    -1.2084   0.1812    -6.6683
## 2|2.25    -0.4470   0.1725    -2.5918
## 2.25|2.5  -0.0569   0.1714    -0.3320
## 2.5|2.75   0.4537   0.1736     2.6131
## 2.75|3     0.9360   0.1801     5.1974
## 3|3.25     1.5737   0.1969     7.9910
## 3.25|3.5   1.8698   0.2085     8.9676
## 3.5|3.75   2.3870   0.2362    10.1055
## 3.75|4     2.7956   0.2669    10.4761
## 4|4.25     3.1435   0.3005    10.4601
## 4.25|4.5   3.4606   0.3385    10.2237
## 4.5|4.75   3.7843   0.3856     9.8134
## 4.75|5     4.0757   0.4364     9.3404
##
## Residual Deviance: 2432.067
## AIC: 2472.067
```

```
summary(model1_share_EN)
```

```
##
## Re-fitting to get Hessian
```

```
## Call:
## polr(formula = as.factor(share_all) ~ as.factor(Feed) + as.factor(valence),
##     data = Data_EN)
##
## Coefficients:
##                       Value Std. Error t value
## as.factor(Feed)2     0.2580     0.2239   1.152
## as.factor(Feed)3    -0.2493     0.2249  -1.108
## as.factor(Feed)4     0.4103     0.2265   1.811
## as.factor(valence)2 -0.1615     0.1609  -1.004
##
## Intercepts:
##     Value   Std. Error t value
## 1|2 -1.2995  0.1894    -6.8605
## 2|3 -0.0114  0.1769    -0.0646
## 3|4  0.6451  0.1797     3.5899
## 4|5  1.4162  0.1915     7.3969
```

```
##
## Residual Deviance: 1544.128
## AIC: 1560.128
```

```r
exp(cbind(OR = coef(model1_cred_EN), confint(model1_cred_EN)))
```

```
## Waiting for profiling to be done...

##
## Re-fitting to get Hessian

##                            OR     2.5 %   97.5 %
## as.factor(Feed)2    1.6293068 1.0514414 2.529411
## as.factor(Feed)3    0.8471983 0.5496425 1.307648
## as.factor(Feed)4    1.5404634 1.0024213 2.380638
## as.factor(valence)2 1.2322066 0.9057622 1.682767
```

```r
exp(cbind(OR = coef(model1_news_EN), confint(model1_news_EN)))
```

```
## Waiting for profiling to be done...
##
## Re-fitting to get Hessian

##                            OR     2.5 %    97.5 %
## as.factor(Feed)2    1.0913650 0.7078932 1.6837522
## as.factor(Feed)3    0.6620673 0.4282892 1.0214895
## as.factor(Feed)4    1.1543404 0.7481560 1.7816590
## as.factor(valence)2 0.5062666 0.3679156 0.6950021
```

```r
exp(cbind(OR = coef(model1_share_EN), confint(model1_share_EN)))
```

```
## Waiting for profiling to be done...
##
## Re-fitting to get Hessian

##                            OR     2.5 %   97.5 %
## as.factor(Feed)2    1.2943328 0.8347292 2.009269
## as.factor(Feed)3    0.7793765 0.5011416 1.210810
## as.factor(Feed)4    1.5071979 0.9672904 2.352341
## as.factor(valence)2 0.8508746 0.6205027 1.166069
```

Chinese dataset:

```r
Data_CN <- na.omit(Data_CN)
Data_CN <- Data_CN %>%
  mutate(across(c("credavg_all", "Feed", "valence"), factor))
model1_cred_CN <- polr(as.factor(credavg_all) ~ as.factor(Feed) + as.factor(valence), data = Data_CN)
model1_news_CN <- polr(as.factor(nwavg_all) ~ as.factor(Feed) + as.factor(valence), data = Data_CN)
model1_share_CN <- polr(as.factor(share_all) ~ as.factor(Feed) + as.factor(valence), data = Data_CN)
summary(model1_cred_CN)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = as.factor(credavg_all) ~ as.factor(Feed) + as.factor(valence),
##     data = Data_CN)
##
## Coefficients:
```

```
##                         Value Std. Error  t value
## as.factor(Feed)2    -0.01966     0.2327 -0.08446
## as.factor(Feed)3    -0.05328     0.2363 -0.22551
## as.factor(Feed)4     0.21878     0.2257  0.96918
## as.factor(valence)2 -0.34925     0.1664 -2.09937
##
## Intercepts:
##           Value    Std. Error t value
## 1|1.25    -4.8587   0.5290    -9.1851
## 1.25|1.5  -4.6331   0.4794    -9.6652
## 1.5|1.75  -4.1559   0.3935   -10.5621
## 1.75|2    -4.0356   0.3754   -10.7497
## 2|2.25    -3.2090   0.2824   -11.3627
## 2.25|2.5  -2.7082   0.2469   -10.9701
## 2.5|2.75  -2.1276   0.2198    -9.6781
## 2.75|3    -1.7585   0.2083    -8.4414
## 3|3.25    -1.2938   0.1976    -6.5477
## 3.25|3.5  -0.9192   0.1921    -4.7857
## 3.5|3.75  -0.3566   0.1876    -1.9007
## 3.75|4     0.2550   0.1868     1.3649
## 4|4.25     0.8545   0.1912     4.4692
## 4.25|4.5   1.2838   0.1982     6.4758
## 4.5|4.75   2.3473   0.2369     9.9074
## 4.75|5     2.8440   0.2701    10.5278
##
## Residual Deviance: 2209.035
## AIC: 2249.035
```

```r
summary(model1_news_CN)
```

```
##
## Re-fitting to get Hessian
## Call:
## polr(formula = as.factor(nwavg_all) ~ as.factor(Feed) + as.factor(valence),
##     data = Data_CN)
##
## Coefficients:
##                         Value Std. Error  t value
## as.factor(Feed)2    -0.02812     0.2370 -0.11869
## as.factor(Feed)3     0.07181     0.2321  0.30939
## as.factor(Feed)4     0.01118     0.2279  0.04907
## as.factor(valence)2  0.35680     0.1674  2.13091
##
## Intercepts:
##           Value    Std. Error t value
## 1|1.25    -5.2279   0.7266    -7.1948
## 1.25|1.5  -4.5305   0.5274    -8.5896
## 1.5|1.75  -4.1203   0.4414    -9.3343
## 1.75|2    -3.7082   0.3732    -9.9371
## 2|2.25    -3.1841   0.3078   -10.3450
## 2.25|2.5  -2.6501   0.2606   -10.1708
## 2.5|2.75  -2.3536   0.2411    -9.7617
## 2.75|3    -1.8742   0.2181    -8.5921
## 3|3.25    -1.1454   0.1986    -5.7668
```

```
## 3.25|3.5  -0.6775   0.1928    -3.5149
## 3.5|3.75   0.0759   0.1891     0.4011
## 3.75|4     0.8692   0.1930     4.5029
## 4|4.25     1.6149   0.2059     7.8436
## 4.25|4.5   2.1844   0.2230     9.7934
## 4.5|4.75   3.3742   0.2952    11.4295
## 4.75|5     4.0891   0.3780    10.8191
##
## Residual Deviance: 2083.05
## AIC: 2123.05
```

```r
summary(model1_share_CN)
```

```
##
## Re-fitting to get Hessian
```

```
## Call:
## polr(formula = as.factor(share_all) ~ as.factor(Feed) + as.factor(valence),
##     data = Data_CN)
##
## Coefficients:
##                       Value Std. Error t value
## as.factor(Feed)2    -0.1261     0.2434 -0.5182
## as.factor(Feed)3    -0.2717     0.2452 -1.1083
## as.factor(Feed)4     0.3752     0.2409  1.5577
## as.factor(valence)2  0.1016     0.1749  0.5807
##
## Intercepts:
##     Value    Std. Error t value
## 1|2  -3.4819   0.3283   -10.6045
## 2|3  -1.7217   0.2146    -8.0222
## 3|4  -0.6894   0.1953    -3.5296
## 4|5   1.2993   0.2037     6.3769
##
## Residual Deviance: 1204.22
## AIC: 1220.22
```

```r
exp(cbind(OR = coef(model1_cred_CN), confint(model1_cred_CN)))
```

```
## Waiting for profiling to be done...
```

```
##
## Re-fitting to get Hessian
```

```
##                            OR     2.5 %    97.5 %
## as.factor(Feed)2    0.9805347 0.6212337 1.5478763
## as.factor(Feed)3    0.9481113 0.5963954 1.5067580
## as.factor(Feed)4    1.2445541 0.7996905 1.9384425
## as.factor(valence)2 0.7052156 0.5085965 0.9765948
```

```r
exp(cbind(OR = coef(model1_news_CN), confint(model1_news_CN)))
```

```
## Waiting for profiling to be done...
##
## Re-fitting to get Hessian
```

```
##                            OR     2.5 %   97.5 %
## as.factor(Feed)2    0.9722674 0.6109265 1.547637
```

```
## as.factor(Feed)3    1.0744468 0.6813655 1.693345
## as.factor(Feed)4    1.0112424 0.6468793 1.581131
## as.factor(valence)2 1.4287557 1.0294872 1.985212
```

```
exp(cbind(OR = coef(model1_share_CN), confint(model1_share_CN)))
```

```
## Waiting for profiling to be done...
##
## Re-fitting to get Hessian
```

```
##                            OR     2.5 %   97.5 %
## as.factor(Feed)2    0.8814858 0.5467776 1.420762
## as.factor(Feed)3    0.7620489 0.4708785 1.232062
## as.factor(Feed)4    1.4553266 0.9082921 2.336804
## as.factor(valence)2 1.1069091 0.7857337 1.560285
```

```
# get coefficients (it's in matrix form)
coefficients <- summary(model1_cred_EN)$coefficients[1:4,]
```

```
##
## Re-fitting to get Hessian
```

```
# calculate p-values
p_value <- (1 - pnorm(abs(coefficients[ ,"t value"]), 0, 1))*2

# bind back to coefficients
(coefficients <- cbind(coefficients, p_value))
```

```
##                          Value Std. Error    t value    p_value
## as.factor(Feed)2     0.4881547  0.2237201  2.1819885 0.02911038
## as.factor(Feed)3    -0.1658205  0.2208879 -0.7506995 0.45283349
## as.factor(Feed)4     0.4320833  0.2204404  1.9600917 0.04998508
## as.factor(valence)2  0.2088065  0.1578954  1.3224358 0.18602306
```

```
# calculate odds ratios
odds_ratio <- exp(coefficients[ ,"Value"])

# combine with coefficient and p_value
(coefficients <- cbind(
  coefficients[ ,c("Value", "p_value")],
  odds_ratio
))
```

```
##                          Value    p_value odds_ratio
## as.factor(Feed)2     0.4881547 0.02911038  1.6293068
## as.factor(Feed)3    -0.1658205 0.45283349  0.8471983
## as.factor(Feed)4     0.4320833 0.04998508  1.5404634
## as.factor(valence)2  0.2088065 0.18602306  1.2322066
```

```
# get coefficients (it's in matrix form)
coefficients <- summary(model1_news_EN)$coefficients[1:4,]
```

```
##
## Re-fitting to get Hessian
```

```
# calculate p-values
p_value <- (1 - pnorm(abs(coefficients[ ,"t value"]), 0, 1))*2

# bind back to coefficients
```

```r
(coefficients <- cbind(coefficients, p_value))
```

```
##                              Value Std. Error    t value      p_value
## as.factor(Feed)2     0.0874292  0.2209137  0.3957619 6.922807e-01
## as.factor(Feed)3    -0.4123881  0.2216091 -1.8608805 6.276106e-02
## as.factor(Feed)4     0.1435291  0.2212228  0.6487988 5.164684e-01
## as.factor(valence)2 -0.6806919  0.1621927 -4.1968104 2.707004e-05
```

```r
# calculate odds ratios
odds_ratio <- exp(coefficients[ ,"Value"])

# combine with coefficient and p_value
(coefficients <- cbind(
  coefficients[ ,c("Value", "p_value")],
  odds_ratio
))
```

```
##                              Value      p_value odds_ratio
## as.factor(Feed)2     0.0874292 6.922807e-01  1.0913650
## as.factor(Feed)3    -0.4123881 6.276106e-02  0.6620673
## as.factor(Feed)4     0.1435291 5.164684e-01  1.1543404
## as.factor(valence)2 -0.6806919 2.707004e-05  0.5062666
```

```r
# get coefficients (it's in matrix form)
coefficients <- summary(model1_share_EN)$coefficients[1:4,]
```

```
##
## Re-fitting to get Hessian
```

```r
# calculate p-values
p_value <- (1 - pnorm(abs(coefficients[ ,"t value"]), 0, 1))*2

# bind back to coefficients
(coefficients <- cbind(coefficients, p_value))
```

```
##                              Value Std. Error   t value    p_value
## as.factor(Feed)2     0.2579954  0.2239389  1.152079 0.24928854
## as.factor(Feed)3    -0.2492610  0.2248866 -1.108385 0.26769537
## as.factor(Feed)4     0.4102522  0.2265470  1.810893 0.07015748
## as.factor(valence)2 -0.1614905  0.1608665 -1.003879 0.31543694
```

```r
# calculate odds ratios
odds_ratio <- exp(coefficients[ ,"Value"])

# combine with coefficient and p_value
(coefficients <- cbind(
  coefficients[ ,c("Value", "p_value")],
  odds_ratio
))
```

```
##                              Value    p_value odds_ratio
## as.factor(Feed)2     0.2579954 0.24928854  1.2943328
## as.factor(Feed)3    -0.2492610 0.26769537  0.7793765
## as.factor(Feed)4     0.4102522 0.07015748  1.5071979
## as.factor(valence)2 -0.1614905 0.31543694  0.8508746
```

```r
# get coefficients (it's in matrix form)
coefficients <- summary(model1_cred_CN)$coefficients[1:4,]
```

```
##
## Re-fitting to get Hessian
```

```r
# calculate p-values
p_value <- (1 - pnorm(abs(coefficients[ ,"t value"]), 0, 1))*2

# bind back to coefficients
(coefficients <- cbind(coefficients, p_value))
```

```
##                           Value Std. Error     t value    p_value
## as.factor(Feed)2    -0.01965725  0.2327401 -0.08446006 0.93269065
## as.factor(Feed)3    -0.05328338  0.2362767 -0.22551265 0.82158048
## as.factor(Feed)4     0.21877732  0.2257338  0.96918289 0.33245395
## as.factor(valence)2 -0.34925173  0.1663606 -2.09936595 0.03578465
```

```r
# calculate odds ratios
odds_ratio <- exp(coefficients[ ,"Value"])

# combine with coefficient and p_value
(coefficients <- cbind(
  coefficients[ ,c("Value", "p_value")],
  odds_ratio
))
```

```
##                           Value    p_value odds_ratio
## as.factor(Feed)2    -0.01965725 0.93269065  0.9805347
## as.factor(Feed)3    -0.05328338 0.82158048  0.9481113
## as.factor(Feed)4     0.21877732 0.33245395  1.2445541
## as.factor(valence)2 -0.34925173 0.03578465  0.7052156
```

```r
# get coefficients (it's in matrix form)
coefficients <- summary(model1_news_CN)$coefficients[1:4,]
```

```
##
## Re-fitting to get Hessian
```

```r
# calculate p-values
p_value <- (1 - pnorm(abs(coefficients[ ,"t value"]), 0, 1))*2

# bind back to coefficients
(coefficients <- cbind(coefficients, p_value))
```

```
##                           Value Std. Error     t value    p_value
## as.factor(Feed)2    -0.02812440  0.2369653 -0.11868574 0.90552433
## as.factor(Feed)3     0.07180589  0.2320867  0.30939260 0.75702290
## as.factor(Feed)4     0.01117969  0.2278540  0.04906516 0.96086737
## as.factor(valence)2  0.35680396  0.1674418  2.13091333 0.03309628
```

```r
# calculate odds ratios
odds_ratio <- exp(coefficients[ ,"Value"])

# combine with coefficient and p_value
(coefficients <- cbind(
  coefficients[ ,c("Value", "p_value")],
```

```r
  odds_ratio
))
```

```
##                              Value    p_value odds_ratio
## as.factor(Feed)2    -0.02812440 0.90552433  0.9722674
## as.factor(Feed)3     0.07180589 0.75702290  1.0744468
## as.factor(Feed)4     0.01117969 0.96086737  1.0112424
## as.factor(valence)2  0.35680396 0.03309628  1.4287557
```

```r
# get coefficients (it's in matrix form)
coefficients <- summary(model1_share_CN)$coefficients[1:4,]
```

```
##
## Re-fitting to get Hessian
```

```r
# calculate p-values
p_value <- (1 - pnorm(abs(coefficients[ ,"t value"]), 0, 1))*2

# bind back to coefficients
(coefficients <- cbind(coefficients, p_value))
```

```
##                             Value Std. Error    t value   p_value
## as.factor(Feed)2    -0.1261464  0.2434238 -0.5182173 0.6043067
## as.factor(Feed)3    -0.2717446  0.2451913 -1.1082962 0.2677339
## as.factor(Feed)4     0.3752304  0.2408913  1.5576751 0.1193103
## as.factor(valence)2  0.1015715  0.1749029  0.5807309 0.5614218
```

```r
# calculate odds ratios
odds_ratio <- exp(coefficients[ ,"Value"])

# combine with coefficient and p_value
(coefficients <- cbind(
  coefficients[ ,c("Value", "p_value")],
  odds_ratio
))
```

```
##                             Value   p_value odds_ratio
## as.factor(Feed)2    -0.1261464 0.6043067  0.8814858
## as.factor(Feed)3    -0.2717446 0.2677339  0.7620489
## as.factor(Feed)4     0.3752304 0.1193103  1.4553266
## as.factor(valence)2  0.1015715 0.5614218  1.1069091
```