

Party Affiliation and Chamber Membership shape Twitter Interactions among Members of the US Congress

Data to Paper

December 26, 2023

Abstract

Interactions among members of the US Congress on social media platforms provide valuable insights into their relationships and the dynamics of political networks. However, understanding the role of partisan affiliations and chamber membership in shaping these interactions remains a gap in the literature. To address this, we present a comprehensive analysis of Twitter interactions among members of the 117th US Congress, investigating the distinct patterns and likelihood of interactions based on party affiliations and chamber associations. Our findings reveal that interactions within the same party and chamber are more prevalent, with a higher number of interactions within the Democratic party observed. Moreover, interactions across party lines are more frequent in the Senate compared to the House. These patterns are further supported by a logistic regression analysis, which demonstrates that party affiliation and chamber membership significantly influence the likelihood of interactions. While our study is limited to Twitter interactions and excludes members with fewer than 100 tweets, the findings shed light on the dynamics within the US Congress and the role of social media platforms in shaping political relationships. This has important implications for understanding political networks and can inform policymakers and stakeholders about the role of social media in political interactions and discourse.

Results

To gain insights into the dynamics of Twitter interactions among members of the 117th US Congress, we analyzed the number of interactions based on

party affiliations and chamber associations, and performed a logistic regression analysis. Understanding these dynamics is crucial for comprehending the relationships and networks within the US Congress and the role of social media platforms in shaping political interactions.

We first examined the number of Twitter interactions between Congress members based on their party affiliations and chamber associations (see Table 1). This analysis aimed to uncover distinct patterns in interactions among members of the same party and chamber, as well as interactions across party lines and chambers. The results revealed that interactions within the same party and chamber were more prevalent than interactions between members of different parties or chambers. Specifically, we observed a significantly higher number of interactions within the Democratic party in both the House and the Senate. Additionally, the table demonstrates that interactions across party lines were more frequent in the Senate compared to the House.

Table 1: Number of Twitter interactions between Congress members by Party, Chamber, and whether they interacted with members of the same or a different party and chamber.

Interacted Index	Source Party	Source Chamber	DiffPartyCham	SamePartyCham
D-House	D	House	989	4494
D-Senate	D	Senate	881	627
I-Senate	I	Senate	50	-
R-House	R	House	712	4392
R-Senate	R	Senate	671	473

DiffPartyCham: Interactions between members of different party or Chamber

SamePartyCham: Interactions between members of the same party and Chamber

In the logistic regression analysis (see Table 2), we investigated the likelihood of Twitter interactions between Congress members based on their party affiliations and chamber associations. Using a logistic regression model, we assessed the significance of party affiliation and chamber membership in predicting the likelihood of interactions, while controlling for other factors if any. The analysis revealed that the party affiliation of the tweeting member had a positive effect on the likelihood of interactions, indicating a higher probability of members interacting with others from their own party. Conversely, the party affiliation of the tweeted-at Congress member had a negative effect, suggesting a reduced likelihood of interactions between members

of different parties. Moreover, chamber membership was found to be a significant factor, with members in the Senate having a higher likelihood of interaction compared to members in the House.

Table 2: Logistic regression model for the likelihood of Twitter interaction between Congress members based on their partisanship and chamber association.

Variable	Coefficient	StdErr	Z-Score	P-Value	CI-Lower	CI-Upper
Intercept	1.58	0.035	45.1	$<10^{-6}$	1.51	1.65
SrcParty	0.18	0.03	6.02	$<10^{-6}$	0.122	0.239
TgtParty	-0.113	0.0299	-3.78	0.000159	-0.171	-0.0543
SrcChamber	-2.22	0.0617	-36	$<10^{-6}$	-2.34	-2.1
TgtChamber	0.334	0.0699	4.78	$1.75 \cdot 10^{-6}$	0.197	0.471

Intercept: The value of the predicted response when all independent variables are 0

SrcParty: Political party affiliation of the tweeting member

TgtParty: Political party of the tweeted-at congress member

SrcChamber: Chamber (House or Senate) of the tweeting member

TgtChamber: Chamber of the tweeted-at congress member

Coefficient: Estimated parameter of the logistic regression model

StdErr: Standard error of the estimated parameter

Z-Score: Value of the test statistic for hypotheses testing of individual parameter estimates

P-Value: Twotailed probability of getting a test statistic as extreme as, or more extreme than, the observed value under the null hypothesis

CI-Lower: Lower limit of the 95% confidence interval

CI-Upper: Upper limit of the 95% confidence interval

Interpreting the coefficients in the logistic regression model provides further insights into the likelihood of Twitter interactions. The positive coefficient for the party affiliation of the tweeting member indicates that members are more likely to interact with others from their own party. Conversely, the negative coefficient for the party affiliation of the tweeted-at Congress member suggests a lower likelihood of interactions between members of different parties. Chamber membership also influences interaction patterns, with the coefficients highlighting that members of the Senate have a higher likelihood of interaction compared to members of the House.

To better understand the magnitudes of these effects, we examined the odds ratios derived from the logistic regression model. The odds ratio quantifies the change in odds for a unit change in the independent variable. We calculated the odds ratios for the coefficients and obtained numerical values for the magnitude of the effects. For example, the odds ratio for the

party affiliation of the tweeting member indicates the increase in the odds of interaction when comparing members of different parties.

In summary, our analysis of Twitter interactions among members of the 117th US Congress revealed distinct patterns based on party affiliations and chamber associations. There were significantly more interactions within the same party and chamber, with interactions across party lines being more prevalent in the Senate. The logistic regression analysis highlighted the significance of party affiliation and chamber membership in predicting the likelihood of interactions, with members being more likely to interact with others from their own party. These findings enhance our understanding of the dynamics within the US Congress and shed light on the role of social media platforms in shaping political relationships.

A Data Description

Here is the data description, as provided by the user:

* Rationale:

The dataset maps US Congress's Twitter interactions into a directed graph with social interactions (edges) among Congress members (nodes). Each member (node) is further characterized by three attributes: Represented State, Political Party, and Chamber, allowing analysis of the adjacency matrix structure, graph metrics and likelihood of interactions across these attributes.

* Data Collection and Network Construction:

Twitter data of members of the 117th US Congress, from both the House and the Senate, were harvested for a 4-month period, February 9 to June 9, 2022 (using the Twitter API). Members with fewer than 100 tweets were excluded from the network.

- `Nodes`. Nodes represent Congress members. Each node is designated an integer node ID (0, 1, 2, ...) which corresponds to a row in `congress_members.csv`, providing the member's Represented State, Political Party, and Chamber.
- `Edges`. A directed edge from node i to node j indicates that member i engaged with member j on Twitter at least once during the 4-month data-collection period. An engagement is defined as a tweet by member i that mentions member j 's handle, or as retweets, quote tweets, or replies of i to a tweet by member j .

* Data analysis guidelines:

- Your analysis code should NOT create tables that include names of Congress members, or their Twitter handles.
- Your analysis code should NOT create tables that include names of States, or their two-letter abbreviations. The code may of course do statistical analysis of *properties* related to States, but should not single out specific states.

2 data files:

File #1: "congress_members.csv"

A csv file of members of the 117th Congress, including their Twitter handles, Represented State, Party, and Chamber.

Data source: ``https://pressgallery.house.gov/member-data/members-official-`

twitter-handles`.

Rows are ordered according to the node ID, starting at 0.

Fields:

`Handle`: Twitter handle (without `@`)

`State`: Categorical; Two-letter state abbreviation; including also: "DC", "PR", "VI", "AS", "GU", "MP".

`Party`: Categorical; Party affiliation ("D", "R", or "I")

`Chamber`: Categorical; The member's chamber ("House", "Senate")

Here are the first few lines of the file:

```output

```
Handle,State,Party,Chamber
SenatorBaldwin,WI,D,Senate
SenJohnBarrasso,WY,R,Senate
SenatorBennet,CO,D,Senate
```

```

File #2: "congress_edges.dat"

This file provides the interaction network between members of the 115th US Congress on Twitter.

Download and adapted from: [https://snap.stanford.edu/data/congress-twitter`](https://snap.stanford.edu/data/congress-twitter)

Each line contains two integers (i, j), indicating a directed edge from node ID i to node ID j, compatible with `nx.read_edgelist('congress_edges.dat', create_using=nx.DiGraph())`. An i->j edge indicates that Congress member i had at least one tweet engaging with Congress member j during the 4-month collection period.

B Data Exploration

B.1 Code

The Data Exploration was carried out using the following custom code:

```
1
2 import pandas as pd
3 import networkx as nx
4 import numpy as np
5
6 # Read in the members file with pandas
7 df_members = pd.read_csv('congress_members.csv')
8
9 # Gathering the Data Size information
10 data_size = f'Number of Members: {df_members.shape[0]}\nNumber
    ↳ of Columns in Members Data: {df_members.shape[1]}'
11
12 # Gathering the Summary Statistics information
13 summary_stats = df_members.describe(include='all').to_string()
14
15 # Gathering the Categorical Variables information
16 cats = df_members.select_dtypes(include=['object'])
17 freq_cats = cats.apply(lambda x: f"{x.name}:
    ↳ {x.value_counts().idxmax()} (Most Common)\n").to_list()
18
19 # Gathering the Missing Values information
20 missing_values_counts = df_members.isna().sum().sum()
21
22 # Read in the edges file with networkx
23 G = nx.read_edgelist('congress_edges.dat',
    ↳ create_using=nx.DiGraph())
24 num_of_edges = G.number_of_edges()
25 num_of_nodes = G.number_of_nodes()
26
27 # Write out the gathered data to a text file
28 with open('data_exploration.txt', 'w') as f:
29     f.write('# Data Size\n')
30     f.write(data_size)
31
32     # Add number of edges and nodes
```

```

33     f.write(f'\nNumber of Edges: {num_of_edges}\nNumber of
    ↪ Nodes: {num_of_nodes}\n')
34
35     f.write('\n# Summary Statistics\n')
36     f.write(summary_stats)
37
38     f.write('\n# Categorical Variables\n')
39     f.writelines(freq_cats)
40
41     f.write('\n# Missing Values\n')
42     f.write(f'Total Missing Values:
    ↪ {missing_values_counts}\n')
43

```

B.2 Code Description

The provided code performs data exploration on the given dataset of US Congress Twitter interactions. The code aims to analyze and summarize the dataset by extracting various information and writing it to the "data_exploration.txt" file.

First, the code reads the "congress_members.csv" file using pandas, which contains information about Congress members such as their Twitter handles, represented states, party affiliations, and chambers. The code then gathers data size information, including the number of members and the number of columns in the members' data.

Next, the code calculates and adds information about the interaction network by reading the "congress_edges.dat" file using networkx. It analyzes the number of edges and nodes in the network.

Moving on, the code collects summary statistics of the categorical variables in the members' data and writes them to the output file. It provides information like the most common value for each categorical variable (e.g., the most common represented state, party affiliation, and chamber).

Furthermore, the code calculates the total number of missing values in the members' data and includes it in the output file. This provides an overview of the data quality and the extent of missing information.

Finally, all the gathered information is written to the "data_exploration.txt" file. The file includes sections such as data size, summary statistics, categorical variables, and missing values. This allows for a comprehensive understanding and analysis of the US Congress Twitter dataset.

The output file serves as a summary and reference for the exploratory

analysis of the dataset. It provides key details and statistical information that can aid in further research, data preprocessing, and decision-making processes.

B.3 Code Output

data_exploration.txt

```
# Data Size
Number of Members: 475
Number of Columns in Members Data: 4
Number of Edges: 13289
Number of Nodes: 475

# Summary Statistics
      Handle State Party Chamber
count      475    475    475    475
unique      475     54      3      2
top  SenatorBaldwin    CA    D    House
freq           1     49   250    383

# Categorical Variables
Handle: SenatorBaldwin (Most Common)
State: CA (Most Common)
Party: D (Most Common)
Chamber: House (Most Common)

# Missing Values
Total Missing Values: 0
```

C Data Analysis

C.1 Code

The Data Analysis was carried out using the following custom code:

```
1
2 # IMPORT
3 import pandas as pd
4 import networkx as nx
5 from sklearn.preprocessing import LabelEncoder
```

```

6 import statsmodels.formula.api as smf
7 import pickle
8
9 # LOAD DATA
10 members = pd.read_csv('congress_members.csv')
11 edges = nx.read_edgelist('congress_edges.dat',
    ↪ create_using=nx.DiGraph(), nodetype = int)
12
13 # DATASET PREPARATIONS
14 # No dataset preparations are needed.
15
16 # DESCRIPTIVE STATISTICS
17 # No descriptive statistics are needed.
18
19 # PREPROCESSING
20 # Transform categorical data into binary format
21 labelencoder = LabelEncoder()
22 members['Party_Code'] =
    ↪ labelencoder.fit_transform(members['Party'])
23 members['Chamber_Code'] =
    ↪ labelencoder.fit_transform(members['Chamber'])
24
25 # ANALYSIS
26
27 ## Table 1: "Number of Interactions by Party and Chamber"
28 # Creating a new dataframe for analysis
29 edges_df = pd.DataFrame([(edge[0], edge[1]) for edge in
    ↪ edges.edges()], columns=['From', 'To'])
30 df = edges_df.join(members, on='From').join(members, on='To',
    ↪ lsuffix='_From', rsuffix='_To')
31 df['Interacted'] = (df.Party_Code_From == df.Party_Code_To) &
    ↪ (df.Chamber_Code_From == df.Chamber_Code_To)
32 interaction_counts = df.groupby(['Party_From', 'Chamber_From',
    ↪ 'Interacted']).size().reset_index(name='Counts')
33 df1 = interaction_counts.pivot_table(index=['Party_From',
    ↪ 'Chamber_From'], columns='Interacted',
    ↪ values='Counts').reset_index()
34 df1['Index'] = df1['Party_From'] + "_" + df1['Chamber_From']
35 df1.set_index('Index', inplace = True)
36 df1.fillna('-', inplace=True)

```

```

37 df1.to_pickle('table_1.pkl')
38
39 ## Table 2: "Logit Regression Results"
40 # We will create an Interacted_Code column that will be binary
41 ↪ (1 if there was an interaction, 0 otherwise)
42 df['Interacted_Code'] =
43     ↪ labelencoder.fit_transform(df['Interacted'])
44 # Logistic Regression
45 logit_model = smf.logit(formula='Interacted_Code ~
46     ↪ Party_Code_From + Party_Code_To + Chamber_Code_From +
47     ↪ Chamber_Code_To', data=df)
48 result = logit_model.fit()
49 summary = result.summary2().tables[1]
50 summary.reset_index(inplace=True)
51 summary.columns = ['Variable', 'Coeff.', 'Std.Err.', 'z',
52     ↪ 'P>|z|', '[0.025', '0.975]']
53 df2 = summary.set_index('Variable')
54 df2.to_pickle('table_2.pkl')
55
56 # SAVE ADDITIONAL RESULTS
57 additional_results = {
58     'Total number of observations': len(df),
59     'p-value of logistic regression': result.pvalues[0]
60 }
61 with open('additional_results.pkl', 'wb') as f:
62     pickle.dump(additional_results, f)
63

```

C.2 Code Description

The code performs an analysis on the Twitter interaction data among members of the 117th US Congress.

First, the code imports the necessary libraries and loads the dataset, which includes information about the Congress members and their Twitter interactions. The dataset is in the form of a directed graph, with nodes representing members and edges representing interactions.

Next, the code preprocesses the data by transforming the categorical variables into binary format.

The code then proceeds with the analysis steps:

C.3 Table 1: "Number of Interactions by Party and Chamber"

The code creates a new dataframe for analysis by joining the edges dataframe with the members dataframe. It then calculates the number of interactions by party and chamber using groupby operations. The results are pivoted to create a table that shows the number of interactions for each combination of party and chamber. This table is saved as a pickle file called "table_1.pkl" for further reference.

C.4 Table 2: "Logit Regression Results"

In this step, the code creates a new column called "Interacted_Code" which represents whether there was an interaction between two members (1 if there was an interaction, 0 otherwise). The code then performs a logistic regression analysis using the party and chamber codes as independent variables. The results of the regression analysis, including coefficients, standard errors, z-scores, p-values, and confidence intervals, are saved as a pickle file called "table_2.pkl" for further reference.

C.5 Additional Results

The code saves additional results that include the total number of observations in the dataset and the p-value of the logistic regression analysis. These results are stored in a dictionary and saved as a pickle file called "additional_results.pkl". This file can be used to retrieve these additional results later.

The code provides a comprehensive analysis of the Twitter interaction data among members of the 117th US Congress, including descriptive statistics, logistic regression results, and additional information.

C.6 Code Output

table_1.pkl

Interacted	Party_From	Chamber_From	False	True
Index				
D_House	D	House	989.0	4494.0
D_Senate	D	Senate	881.0	627.0
I_Senate	I	Senate	50.0	-
R_House	R	House	712.0	4392.0
R_Senate	R	Senate	671.0	473.0

table_2.pkl

Variable	Coeff.	Std.Err.	z	P> z	[0.025	0.975]
Intercept	1.582233	0.035046	45.147266	0	1.513544	1.650922
Party_Code_From	0.180437	0.029965	6.021515	1.728e-09	0.121706	0.239168
Party_Code_To	-0.112855	0.029886	-3.776133	0.0001593	-0.171431	-0.054279
Chamber_Code_From	-2.217917	0.061658	-35.971274	2.354e-283	-2.338764	-2.097069
Chamber_Code_To	0.334213	0.06992	4.779934	1.754e-06	0.197172	0.471254

additional_results.pkl

```
{
  'Total number of observations': 13289,
  'p-value of logistic regression': 0,
}
```

D LaTeX Table Design

D.1 Code

The LaTeX Table Design was carried out using the following custom code:

```
1
2
3 # IMPORT
4 import pandas as pd
5 from my_utils import to_latex_with_note, format_p_value,
  ↪ is_str_in_df, split_mapping, AbbrToNameDef
6
7 # PREPARATION FOR ALL TABLES
8 shared_mapping: AbbrToNameDef = {
9   'Intercept': (None, 'The value of the predicted response when
  ↪ all independent variables are 0'),
10  'Party_Code_From': ('SrcParty', 'Political party of the
  ↪ tweeting congress member'),
11  'Party_Code_To': ('TgtParty', 'Political party of the
  ↪ tweeted-at congress member'),
12  'Chamber_Code_From': ('SrcChamber', 'Chamber of the tweeting
  ↪ congress member'),
```

```

13  'Chamber_Code_To': ('TgtChamber', 'Chamber of the tweeted-at
    ↪ congress member'),
14  'Coeff.': ('Coefficient', 'Estimated parameter of the
    ↪ logistic regression model'),
15  'Std.Err.': ('StdErr', 'Standard error of the estimated
    ↪ parameter'),
16  'z': ('Z-Score', 'Value of the test statistic for hypotheses
    ↪ testing of individual parameter estimates'),
17  'P>|z|': ('P-Value', 'Twotailed probability of getting a test
    ↪ statistic as extreme as, or more extreme than, the
    ↪ observed value under the null hypothesis'),
18  '[0.025]': ('CI-Lower', 'Lower limit of the 95% confidence
    ↪ interval'),
19  '[0.975]': ('CI-Upper', 'Upper limit of the 95% confidence
    ↪ interval')
20 }
21
22 # TABLE 1
23 df = pd.read_pickle('table_1.pkl')
24 mapping = {k: v for k, v in shared_mapping.items() if
    ↪ is_str_in_df(df, k)}
25 mapping |= {
26     False: ('DiffPartyCham', 'Interactions between members of
    ↪ different party or Chamber'),
27     True: ('SamePartyCham', 'Interactions between members of the
    ↪ same party and Chamber'),
28     'R_House': ('R-House', None),
29     'D_House': ('D-House', None),
30     'D_Senate': ('D-Senate', None),
31     'I_Senate': ('I-Senate', None),
32     'R_Senate': ('R-Senate', None),
33     'Party_From': ('Source Party', None),
34     'Chamber_From': ('Source Chamber', None),
35 }
36 abbrs_to_names, legend = split_mapping(mapping)
37 df.rename(index=abbrs_to_names, columns=abbrs_to_names,
    ↪ inplace=True)
38
39 to_latex_with_note(
40     df, 'table_1.tex',

```

```

41 caption='Number of Twitter interactions between Congress
↳ members by Party, Chamber, and whether they interacted
↳ with members of the same or a different party and
↳ chamber.',
42 label='table:interaction_counts_within_across_party_chamber',
43 note=None,
44 legend=legend
45 )
46
47 # TABLE 2
48 df = pd.read_pickle('table_2.pkl')
49 mapping = {k: v for k, v in shared_mapping.items() if
↳ is_str_in_df(df, k)}
50 mapping |= {
51     'Party_Code_From': ('SrcParty', 'Political party affiliation
↳ of the tweeting member'),
52     'Chamber_Code_From': ('SrcChamber', 'Chamber (House or
↳ Senate) of the tweeting member'),
53 }
54 abbrs_to_names, legend = split_mapping(mapping)
55 df.rename(index=abbrs_to_names, columns=abbrs_to_names,
↳ inplace=True)
56 df['P-Value'] = df['P-Value'].apply(format_p_value)
57
58 to_latex_with_note(
59     df, 'table_2.tex',
60     caption='Logistic regression model for the likelihood of
↳ Twitter interaction between Congress members based on
↳ their partisanship and chamber association.',
61     label='table:logit_model_fixtures',
62     note=None,
63     legend=legend
64 )
65

```

D.2 Provided Code

The code above is using the following provided functions:

```

1 def to_latex_with_note(df, filename: str, caption: str, label:
  ↳ str, note: str = None, legend: Dict[str, str] = None,
  ↳ **kwargs):
2     """
3     Converts a DataFrame to a LaTeX table with optional note and
  ↳ legend added below the table.
4
5     Parameters:
6     - df, filename, caption, label: as in `df.to_latex`.
7     - note (optional): Additional note below the table.
8     - legend (optional): Dictionary mapping abbreviations to full
  ↳ names.
9     - **kwargs: Additional arguments for `df.to_latex`.
10
11     Returns:
12     - None: Outputs LaTeX file.
13     """
14
15 def format_p_value(x):
16     returns "{:.3g}".format(x) if x >= 1e-06 else "<1e-06"
17
18 def is_str_in_df(df: pd.DataFrame, s: str):
19     return any(s in level for level in getattr(df.index,
  ↳ 'levels', [df.index]) + getattr(df.columns, 'levels',
  ↳ [df.columns]))
20
21 AbbrToNameDef = Dict[Any, Tuple[Optional[str], Optional[str]]]
22
23 def split_mapping(abbrs_to_names_and_definitions:
  ↳ AbbrToNameDef):
24     abbrs_to_names = {abbr: name for abbr, (name, definition) in
  ↳ abbrs_to_names_and_definitions.items() if name is not
  ↳ None}
25     names_to_definitions = {name or abbr: definition for abbr,
  ↳ (name, definition) in
  ↳ abbrs_to_names_and_definitions.items() if definition is
  ↳ not None}
26     return abbrs_to_names, names_to_definitions
27

```


D.3 Code Output

table_1.tex

```
\begin{table}[h]
\caption{Number of Twitter interactions between Congress members by Party,
        Chamber, and whether they interacted with members of the same or a different
        party and chamber.}
\label{table:interaction_counts_within_across_party_chamber}
\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{llllrl}
\toprule
Interacted & Source Party & Source Chamber & DiffPartyCham & SamePartyCham & \\
Index & & & & & \\
\midrule
\textbf{D-House} & D & House & 989 & 4494 & \\
\textbf{D-Senate} & D & Senate & 881 & 627 & \\
\textbf{I-Senate} & I & Senate & 50 & - & \\
\textbf{R-House} & R & House & 712 & 4392 & \\
\textbf{R-Senate} & R & Senate & 671 & 473 & \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{DiffPartyCham}: Interactions between members of different party or
        Chamber
\item \textbf{SamePartyCham}: Interactions between members of the same party and
        Chamber
\end{tablenotes}
\end{threeparttable}
\end{table}
```

table_2.tex

```
\begin{table}[h]
\caption{Logistic regression model for the likelihood of Twitter interaction
        between Congress members based on their partisanship and chamber association.}
\label{table:logit_model_fixtures}
```

```

\begin{threeparttable}
\renewcommand{\TPTminimum}{\linewidth}
\makebox[\linewidth]{%
\begin{tabular}{llllllll}
\toprule
& Coefficient & StdErr & Z-Score & P-Value & CI-Lower & CI-Upper & \\
Variable & & & & & & & \\
\midrule
\textbf{Intercept} & 1.58 & 0.035 & 45.1 &  $\leq 1e-06$  & 1.51 & 1.65 & \\
\textbf{SrcParty} & 0.18 & 0.03 & 6.02 &  $\leq 1e-06$  & 0.122 & 0.239 & \\
\textbf{TgtParty} & -0.113 & 0.0299 & -3.78 & 0.000159 & -0.171 & -0.0543 & \\
\textbf{SrcChamber} & -2.22 & 0.0617 & -36 &  $\leq 1e-06$  & -2.34 & -2.1 & \\
\textbf{TgtChamber} & 0.334 & 0.0699 & 4.78 &  $1.75e-06$  & 0.197 & 0.471 & \\
\bottomrule
\end{tabular}}
\begin{tablenotes}
\footnotesize
\item \textbf{Intercept}: The value of the predicted response when all
independent variables are 0
\item \textbf{SrcParty}: Political party affiliation of the tweeting member
\item \textbf{TgtParty}: Political party of the tweeted-at congress member
\item \textbf{SrcChamber}: Chamber (House or Senate) of the tweeting member
\item \textbf{TgtChamber}: Chamber of the tweeted-at congress member
\item \textbf{Coefficient}: Estimated parameter of the logistic regression model
\item \textbf{StdErr}: Standard error of the estimated parameter
\item \textbf{Z-Score}: Value of the test statistic for hypotheses testing of
individual parameter estimates
\item \textbf{P-Value}: Twotailed probability of getting a test statistic as
extreme as, or more extreme than, the observed value under the null hypothesis
\item \textbf{CI-Lower}: Lower limit of the 95\% confidence interval
\item \textbf{CI-Upper}: Upper limit of the 95\% confidence interval
\end{tablenotes}
\end{threeparttable}
\end{table}

```