Results

Notes:
- Some of this is already in or was based on the blogpost/interface code. Hit show to see code. I switch between R and Python - Some of this won't make it to the paper. You can probably skip preprocessing unless you want to check certain things, example: did we make sure to remove judgments based on X condition

Preprocessing

Importing, filtering, and adding columns

We have 3 sets of data from the interface:

```
import pandas as pd
import numpy as np
import altair as alt
import math as math
import matplotlib.pyplot as plt
import re
pd.options.mode.chained_assignment = None  # default='warn'

# Load summaries that can be downloaded from the interface
data_path = "/Users/bila/git/for-debate/debate/save/official/summaries/"
debates = pd.read_csv(data_path + "debates.csv", keep_default_na=True)
sessions = pd.read_csv(data_path + "sessions.csv", keep_default_na=True)
turns = pd.read_csv(data_path + "turns.csv", keep_default_na=True)
print(f' {debates.shape} - Debates') ;
```

```
##  (631, 29) - Debates
```

```
print(f'{sessions.shape} - Sessions, which has multiple rows (of participants) for each debate') ;
```

```
## (1869, 46) - Sessions, which has multiple rows (of participants) for each debate
```

```
print(f'{turns.shape} - and Turns, which has multiple rows (of participant turns) for each debate')
```

```
## (6259, 15) - and Turns, which has multiple rows (of participant turns) for each debate
```

```
# Only include debates within a given period
debates["Start time"] = pd.to_datetime(debates["Start time"], unit="ms")
debates["End time"] = pd.to_datetime(debates["End time"], unit="ms")
debates["Last modified time"] = pd.to_datetime(debates["Last modified time"], unit="ms")
debates = debates[
```

```
    (debates["Start time"] > pd.to_datetime("10/02/23", format="%d/%m/%y")) &
    (debates["End time"] < pd.to_datetime("01/09/23", format="%d/%m/%y"))
]
### for filtering to when we had AI debates: 16/07/23
# Filter sessions & turns to only the selected debates
sessions = sessions.merge(debates[["Room name"]], how="inner", on="Room name")
turns = turns.merge(debates[["Room name"]], how="inner", on="Room name")
print(f'We have {len(debates)} debates when filtering out the initial pilots last fall')
```

```
## We have 582 debates when filtering out the initial pilots last fall
```

```
# Secondary analysis: Question Difficulty
# Create new columns with bin labels
debates['Untimed annotator context bins'] = pd.cut(debates['Untimed annotator context'].round(), bins=[
#print(debates['Untimed annotator context'].round().value_counts()) #check
#print(debates['Untimed annotator context bins'].value_counts()) #check
debates['Speed annotator accuracy bins'] = pd.cut(debates['Speed annotator accuracy'].round(1), bins=[0
## respectively, those speed annotator accuracies probably mean 0 right, 1 right, 2 right
#print(debates['Speed annotator accuracy'].round(1).value_counts().sort_index()) #check #0.5 acc?
#print(debates['Speed annotator accuracy bins'].value_counts().sort_index()) #check

debates['Final_Accuracy'] = debates['Final probability correct'] > 0.5

print(f'Average accuracy per context required by question:\n{debates.groupby("Untimed annotator context
```

```
## Average accuracy per context required by question:
##                                 Proportion_True  Total_Count
## Untimed annotator context bins
## 1                                      0.781250           64
## 2                                      0.711382          246
## 3                                      0.702857          175
## 4                                      0.639175           97
## Overall accuracy goes down the more context is required
```

```
print(f'Average accuracy per difficulty based on speed annotator accuracy:\n{debates.groupby("Speed ann
```

```
## Average accuracy per difficulty based on speed annotator accuracy:
##                                Proportion_True  Total_Count
## Speed annotator accuracy bins
## 0                                     0.728682          129
## 0.1                                        NaN            0
## 0.2                                   0.697509          281
## 0.3                                   0.666667            3
## 0.4                                   0.698795          166
## 0.5                                   0.666667            3
## Hm, this seems less likely to be a good indicator of question difficulty
```

```
# Determine settings for each row
def setups(row):
    if 'GPT-4' in (row['Honest debater'], row['Dishonest debater']):
```

```
        if row['Is single debater']:
            return "AI Consultancy " + ("Honest" if row['Has honest debater'] else "Dishonest")
        else:
            return "AI Debate"
    else:
        if row['Is single debater']:
            return "Human Consultancy " + ("Honest" if row['Has honest debater'] else "Dishonest")
        else:
            return "Human Debate"

debates['Setting'] = debates.apply(setups, axis=1)
# Agregate settings - the 4 that we normally talk about:
debates['Final_Setting'] = debates['Setting'].str.replace(' Honest', '').str.replace(' Dishonest', '')
```

Merging, filtering for judgments

```
# Merge sessions with debates, so we have each judge's final probability correct and the debate's metad
source = sessions.merge(
        debates[["Room name", "Debater A","Debater B","Honest debater", "Dishonest debater",
                 "Is single debater", 'Has honest debater',
                 "Final_Setting", "Setting",
                 "Question", "Article ID", "Story length",
                 "Speed annotator accuracy bins","Untimed annotator context bins",
                 "Speed annotator accuracy","Untimed annotator context", "Is offline",
                 'End time', 'Last modified time']],
        how="left",
        on="Room name",
    )
print(f'After merging debates with sessions, we have the following participant counts for those debates
```

```
## After merging debates with sessions, we have the following participant counts for those debates:
## Judge            548
## Debater B        486
## Debater A        457
## Offline Judge    232
## Name: Role, dtype: int64
```

```
#[source['Is over'] == True] to check for completed online/offline debates

# Filter out incomplete judgments
judgments = source[source['Final probability correct'].notnull()]
print(f'After filtering to judges that have finalized their judgment, we have the following judgments p
```

```
## After filtering to judges that have finalized their judgment, we have the following judgments per rol
## Judge            507
## Offline Judge    223
## Name: Role, dtype: int64
## for a total of 730 judgments.
```

```
print(f'Of those judgments, we have this much for each setting (not consolidating honest - dishonest co
```

```
## Of those judgments, we have this much for each setting (not consolidating honest - dishonest consulta
## Human Debate                     419
## AI Debate                         92
## Human Consultancy Dishonest       69
## AI Consultancy Honest             56
## Human Consultancy Honest          54
## AI Consultancy Dishonest          40
## Name: Setting, dtype: int64
```

```
judgments['Final_Accuracy'] = judgments['Final probability correct'] > 0.5

print(f'Of those judgments, we have this much for each setting (aggregated):\n{judgments.groupby("Final_
```

```
## Of those judgments, we have this much for each setting (aggregated):
##                      Proportion_True  Total_Count
## Final_Setting
## AI Consultancy              0.802083           96
## AI Debate                   0.782609           92
## Human Consultancy           0.707317          123
## Human Debate                0.878282          419
```

```
# Remove judges who see the story more than once
judgments['base_room_name'] = judgments['Room name'].str.extract('(.*)\d+$', expand=False).fillna(judgme
judgments = judgments.sort_values(by=['base_room_name','End time']).groupby(['Participant', 'base_room_n

print(f'1. We then filter to judgments where the judge has only seen a story once, and now we have this
```

```
## 1. We then filter to judgments where the judge has only seen a story once, and now we have this much
##                      Proportion_True  Total_Count
## Final_Setting
## AI Consultancy              0.802083           96
## AI Debate                   0.782609           92
## Human Consultancy           0.707317          123
## Human Debate                0.869452          383
```

```
# Filter to online judges only
judgments_online = judgments[judgments["Role"] == "Judge"]
print(f'2. We\'ll make a copy of the online judgments only leaving us with the following judgments:\n{ju
```

```
## 2. We'll make a copy of the online judgments only leaving us with the following judgments:
##                      Proportion_True  Total_Count
## Final_Setting
## AI Consultancy              0.797872           94
## AI Debate                   0.791209           91
## Human Consultancy           0.709091          110
## Human Debate                0.865979          194
```

```
judgments_online = judgments_online[judgments_online['Untimed annotator context bins'].isin(['2', '3', 

print(f'3. We then filter to judgments which require more than a sentence or two, and now we have this m
```

```
## 3. We then filter to judgments which require more than a sentence or two, and now we have this much 
##                      Proportion_True  Total_Count
## Final_Setting
## AI Consultancy              0.806452           93
## AI Debate                   0.781609           87
## Human Consultancy           0.700935          107
## Human Debate                0.844156          154
## This is where debate accuracy drops
```

```
pd.set_option('display.max_columns', None)
total_counts_for_setting = judgments_online.groupby('Final_Setting').size()
result = judgments_online.groupby(["Final_Setting", "Untimed annotator context bins"], observed=False).a
    Proportion_True=pd.NamedAgg(column='Final_Accuracy', aggfunc=lambda x: x.mean()),
    Count=pd.NamedAgg(column='Final_Accuracy', aggfunc='size'),
    Proportion_Count=pd.NamedAgg(column='Final_Setting', aggfunc=lambda x: len(x) / total_counts_for_set
)
print(f'Are the difficult questions equally enough distributed amongst settings?:\n{result}')
```

```
## Are the difficult questions equally enough distributed amongst settings?:
##                                                 Proportion_True  Count  \
## Final_Setting     Untimed annotator context bins
## AI Consultancy    1                                         NaN      0
##                   2                                    0.823529     51
##                   3                                    0.826087     23
##                   4                                    0.736842     19
## AI Debate         1                                         NaN      0
##                   2                                    0.777778     45
##                   3                                    0.772727     22
##                   4                                    0.800000     20
## Human Consultancy 1                                         NaN      0
##                   2                                    0.634146     41
##                   3                                    0.708333     48
##                   4                                    0.833333     18
## Human Debate      1                                         NaN      0
##                   2                                    0.890411     73
##                   3                                    0.816667     60
##                   4                                    0.761905     21
##
##                                                 Proportion_Count
## Final_Setting     Untimed annotator context bins
## AI Consultancy    1                                          NaN
##                   2                                     0.548387
##                   3                                     0.247312
##                   4                                     0.204301
## AI Debate         1                                          NaN
##                   2                                     0.517241
##                   3                                     0.252874
##                   4                                     0.229885
```

```
## Human Consultancy 1                                             NaN
##                    2                                        0.383178
##                    3                                        0.448598
##                    4                                        0.168224
## Human Debate       1                                             NaN
##                    2                                        0.474026
##                    3                                        0.389610
##                    4                                        0.136364
```

```
pd.reset_option('display.max_columns')
```

So question difficulty isn't perfectly balanced... but consultancies have a different relationship with question difficulty anyway? **need a second opinion** We might at least want to ratio it better for AI settings...

Trying to balance the data

1. Balancing honest & dishonest consultancies
2. Question weights

Balancing honest & dishonest consultancies

```
def balance_consultancies(df, sample_setting, random_state):
    """
    Sample distinct questions, then use common questions, ensure equal counts.
    """
    consult_df = df[df['Setting'].str.contains(sample_setting, na=False)]
    honest_df = consult_df[consult_df['Setting'].str.contains('Honest')]
    dishonest_df = consult_df[consult_df['Setting'].str.contains('Dishonest')]
    sample_column_name = f'{sample_setting} Sample'
    df[sample_column_name] = False
    # Separate into distinct and common questions
    # First, let's extract the combinations of 'Article ID' and 'Question' for both honest and dishones
    honest_combinations = set(honest_df[['Article ID', 'Question']].itertuples(index=False, name=None))
    dishonest_combinations = set(dishonest_df[['Article ID', 'Question']].itertuples(index=False, name=
    # Identifying the common and distinct combinations
    common_combinations = honest_combinations.intersection(dishonest_combinations)
    distinct_honest_combinations = honest_combinations - common_combinations
    distinct_dishonest_combinations = dishonest_combinations - common_combinations
    # Filtering the original dataframes based on these combinations to get distinct and common datafram
    common_honest_df = honest_df[honest_df.set_index(['Article ID', 'Question']).index.isin(common_comb
    common_dishonest_df = dishonest_df[dishonest_df.set_index(['Article ID', 'Question']).index.isin(co
    distinct_honest_df = honest_df[honest_df.set_index(['Article ID', 'Question']).index.isin(distinct_
    distinct_dishonest_df = dishonest_df[dishonest_df.set_index(['Article ID', 'Question']).index.isin(
    def extract_correct_index(sample_df):
        if isinstance(sample_df.index, pd.MultiIndex):
            return sample_df.index.get_level_values(2)
        else:
            return sample_df.index
    # Get distinct consultancies
    sample_size = min(len(distinct_honest_df), len(distinct_dishonest_df))
    honest_sample = distinct_honest_df.sample(sample_size, random_state=random_state)
```

```python
    dishonest_sample = distinct_dishonest_df.sample(sample_size, random_state=random_state)
    df.loc[extract_correct_index(honest_sample), sample_column_name] = True
    df.loc[extract_correct_index(dishonest_sample), sample_column_name] = True
    # Drop sampled questions from distinct dataframes
    honest_remove_distinct = set(honest_sample[['Article ID', 'Question']].itertuples(index=False, name=
    dishonest_remove_distinct = set(dishonest_sample[['Article ID', 'Question']].itertuples(index=False
    distinct_honest_df = distinct_honest_df[~distinct_honest_df.index.isin(honest_sample.index)]
    distinct_dishonest_df = distinct_dishonest_df[~distinct_dishonest_df.index.isin(dishonest_sample.in
    honest_distinct_remaining = len(distinct_honest_df)
    dishonest_distinct_remaining = len(distinct_dishonest_df)
    # Sample from remaining distinct questions, using common questions for the other (bigger count) set
    if honest_distinct_remaining > dishonest_distinct_remaining:
        sample_size = min(honest_distinct_remaining, len(common_dishonest_df))
        honest_sample = distinct_honest_df.sample(sample_size, random_state=random_state)
        dishonest_sample = common_dishonest_df.sample(sample_size, random_state=random_state)
        df.loc[extract_correct_index(dishonest_sample), sample_column_name] = True
        df.loc[extract_correct_index(honest_sample), sample_column_name] = True
        dishonest_remove_common = set(dishonest_sample[['Article ID', 'Question']].itertuples(index=Fals
        common_dishonest_df = common_dishonest_df[~common_dishonest_df.index.isin(dishonest_sample.index
        common_honest_df = common_honest_df[~common_honest_df.index.isin(honest_sample.index)]
    else:
        sample_size = min(dishonest_distinct_remaining, len(common_honest_df))
        honest_sample = common_honest_df.sample(sample_size, random_state=random_state)
        dishonest_sample = distinct_dishonest_df.sample(sample_size, random_state=random_state)
        df.loc[extract_correct_index(dishonest_sample), sample_column_name] = True
        df.loc[extract_correct_index(honest_sample), sample_column_name] = True
        honest_remove_common = set(honest_sample[['Article ID', 'Question']].itertuples(index=False, nam
        common_dishonest_df = common_dishonest_df[~common_dishonest_df.index.isin(dishonest_sample.index
        common_honest_df = common_honest_df[~common_honest_df.index.isin(honest_sample.index)]
    # Remaining independent samples from common_honest_df
    if len(common_honest_df) > 0 or len(common_dishonest_df) > 0:
        sample_size = min(len(common_honest_df), len(common_dishonest_df))
        honest_sample = common_honest_df.sample(sample_size, random_state=random_state)
        dishonest_sample = common_dishonest_df.sample(sample_size, random_state=random_state)
        df.loc[extract_correct_index(honest_sample), sample_column_name] = True
        df.loc[extract_correct_index(dishonest_sample), sample_column_name] = True
    return df

# Run the sampling to balance the consultancies
judgments_online = balance_consultancies(judgments_online, 'Human Consultancy', random_state = 12345)
judgments_online = balance_consultancies(judgments_online, 'AI Consultancy', random_state = 12345)
# Create one sample column for easier indexing, create mask
#sample_columns = [col for col in judgments_online.columns if 'Sample' in col]
#judgments_online['Sample'] = judgments_online[sample_columns].any(axis=1)
#consultancy_balanced = (~judgments_online['Setting'].str.contains('Consultancy', case=False, na=False)

#print(f'Accuracy after balancing consultancies:\n{judgments_online[consultancy_balanced].groupby(["Fin


#from statsmodels.stats.proportion import proportions_ztest

#def run_experiment(judgments_online):
#    judgments_online['Sample'] = False
```

```
#    judgments_online = balance_consultancies(judgments_online, 'Human Consultancy')
#    judgments_online = balance_consultancies(judgments_online, 'AI Consultancy')
#    sample_columns = [col for col in judgments_online.columns if 'Sample' in col]
#    judgments_online['Sample'] = judgments_online[sample_columns].any(axis=1)
#    consultancy_balanced = (~judgments_online['Setting'].str.contains('Consultancy', case=False, na=Fa
#    result = judgments_online[consultancy_balanced].groupby(["Final_Setting"])["Final_Accuracy"].agg(P
#    return result

# Number of iterations
#num_iterations = 1000

# Store results from each iteration
#results = []
#p_vals = []
# Run the experiment multiple times
#for _ in range(num_iterations):
#    result = run_experiment(judgments_online.copy())  # Use a copy to ensure original data remains unc
#    results.append(result)
#    # Run the proportions test
#    group_human_debate = result.loc['Human Debate']
#    group_human_consultancy = result.loc['Human Consultancy']
#    count = [group_human_debate.Proportion_True * group_human_debate.Total_Count, group_human_consulta
#    nobs = [group_human_debate.Total_Count, group_human_consultancy.Total_Count]
#    z_stat, p_val = proportions_ztest(count, nobs)
#    p_vals.append(p_val)

# Calculate the average of the results
#average_result = pd.concat(results).groupby(level=0).mean()

#print(f'\nAverage accuracy after {num_iterations} iterations:\n{average_result}')

#print(f'pval mean: {np.mean(p_vals)}')
```

Balance debates? (not actually used)

```python
def balance_debates(df, sample_setting, random_state):
    debates_df = df[df['Setting'].str.contains(sample_setting, na=False)]
    sample_column_name = f'{sample_setting} Sample'
    df[sample_column_name] = False
    def extract_correct_index(sample_df):
        if isinstance(sample_df.index, pd.MultiIndex):
            return sample_df.index.get_level_values(2)
        else:
            return sample_df.index
    # Get distinct consultancies
    sample_size = len(debates_df.groupby(['Question', 'Article ID']))
    sample_debates = debates_df.groupby(['Question', 'Article ID']).apply(lambda x: x.sample(1, random_
    df.loc[extract_correct_index(sample_debates), sample_column_name] = True
    return df

# Run the sampling to balance the consultancies
```

```
judgments_online = balance_debates(judgments_online, 'Human Debate', random_state = 123)
judgments_online = balance_debates(judgments_online, 'AI Debate', random_state = 123)
```

Question weights

```
# Create one sample column for easier indexing, create mask
sample_columns = [col for col in judgments_online.columns if 'Sample' in col]
consultancy_sample_columns = [col for col in judgments_online.columns if 'Consultancy Sample' in col]
judgments_online['Sample'] = judgments_online[sample_columns].any(axis=1)
judgments_online['Consultancy Sample'] = judgments_online[consultancy_sample_columns].any(axis=1)
consultancy_balanced = (~judgments_online['Setting'].str.contains('Consultancy', case=False, na=False))

print(f'Accuracy per setting (aggregated) after balancing:\n{judgments_online[consultancy_balanced].grou
```

```
## Accuracy per setting (aggregated) after balancing:
##                     Proportion_True  Total_Count
## Final_Setting
## AI Consultancy             0.828947           76
## AI Debate                  0.781609           87
## Human Consultancy          0.718750           96
## Human Debate               0.844156          154
## Accuracies remain pretty similar
```

```
def question_weights(data, columns, weight_column_name, consultancy_sample=None, debate_sample=None):
    # 0. Make a copy of the original data for weight calculations
    working_data = data.copy()
    # 0.1. Custom filtering based on the 'Setting' column
    consultancy_condition = working_data['Setting'].str.contains('Consultancy', case=False, na=False)
    debate_condition = ~consultancy_condition
    if consultancy_sample is not None:
        consultancy_condition &= (working_data['Sample'] == consultancy_sample)
    if debate_sample is not None: # uncomment if we want to sample debates
        debate_condition &= (working_data['Sample'] == debate_sample)
    combined_mask = consultancy_condition | debate_condition
    working_data = working_data[combined_mask]
    # 1. Calculate the frequency of each question in the dataset
    question_frequency = working_data.groupby(columns).size()
    # 2. Invert the frequency to get the weight for each question
    question_weights = 1 / question_frequency
    # 3. Normalize the weights
    #question_weights = question_weights / question_weights.sum() * len(question_weights)
    # 4. Assign the calculated weights to the original data and fill missing values with 0
    data.loc[combined_mask, weight_column_name] = data[combined_mask].set_index(columns).index.map(ques
    data[weight_column_name].fillna(0, inplace=True)
    return data

judgments_online = question_weights(
    data=judgments_online,
    columns=['Article ID', 'Question'],
```

```
    weight_column_name='initial_question_weights'
)
judgments_online = question_weights(
    data=judgments_online,
    columns=['Article ID', 'Question', 'Final_Setting'],
    weight_column_name='initial_question_weights_grouped_setting'
)
```

```
def print_weight_summary_by_setting(df, weight_column, consultancy_sample=None):
    consultancy_condition = df['Setting'].str.contains('Consultancy', case=False, na=False)
    if consultancy_sample is not None:
        consultancy_condition &= (df['Consultancy Sample'] == consultancy_sample)
    for setting in sorted(df['Setting'].unique()):
        total_weight = df[df['Setting'] == setting][weight_column].sum()
        print(f"Total {weight_column} for {setting}: {total_weight:.2f}")
    print("\n")

print('Unsampled consultancies/debates (initial) weights, by group setting')
```

```
## Unsampled consultancies/debates (initial) weights, by group setting
```

```
print_weight_summary_by_setting(judgments_online, 'initial_question_weights_grouped_setting')
```

```
## Total initial_question_weights_grouped_setting for AI Consultancy Dishonest: 32.50
## Total initial_question_weights_grouped_setting for AI Consultancy Honest: 49.50
## Total initial_question_weights_grouped_setting for AI Debate: 75.00
## Total initial_question_weights_grouped_setting for Human Consultancy Dishonest: 34.67
## Total initial_question_weights_grouped_setting for Human Consultancy Honest: 26.33
## Total initial_question_weights_grouped_setting for Human Debate: 107.00
```

```
# Recalculate weights for balanced consultancies, all debates
judgments_online = question_weights(
    data=judgments_online,
    columns=['Article ID', 'Question'],
    weight_column_name='sampled_consultancies_all_debates_weights',
    consultancy_sample=True
)
judgments_online = question_weights(
    data=judgments_online,
    columns=['Article ID', 'Question', 'Setting'],
    weight_column_name='sampled_consultancies_all_debates_weights_setting',
    consultancy_sample=True
)
judgments_online = question_weights(
    data=judgments_online,
    columns=['Article ID', 'Question', 'Final_Setting'],
    weight_column_name='sampled_consultancies_all_debates_weights_grouped_setting',
    consultancy_sample=True
)
print('Consultancy balanced weights, not grouped - (not balanced, would have to change balancing functi
```

```
## Consultancy balanced weights, not grouped - (not balanced, would have to change balancing function..
```

```
print_weight_summary_by_setting(judgments_online[consultancy_balanced], 'sampled_consultancies_all_deba
```

```
## Total sampled_consultancies_all_debates_weights for AI Consultancy Dishonest: 27.82
## Total sampled_consultancies_all_debates_weights for AI Consultancy Honest: 36.45
## Total sampled_consultancies_all_debates_weights for AI Debate: 66.47
## Total sampled_consultancies_all_debates_weights for Human Consultancy Dishonest: 16.60
## Total sampled_consultancies_all_debates_weights for Human Consultancy Honest: 17.37
## Total sampled_consultancies_all_debates_weights for Human Debate: 81.30
```

```
print('Consultancy balanced weights, grouped by Setting - see that the consultancies are balanced betwe
```

```
## Consultancy balanced weights, grouped by Setting - see that the consultancies are balanced between th
```

```
print_weight_summary_by_setting(judgments_online[consultancy_balanced], 'sampled_consultancies_all_deba
```

```
## Total sampled_consultancies_all_debates_weights_setting for AI Consultancy Dishonest: 38.00
## Total sampled_consultancies_all_debates_weights_setting for AI Consultancy Honest: 38.00
## Total sampled_consultancies_all_debates_weights_setting for AI Debate: 75.00
## Total sampled_consultancies_all_debates_weights_setting for Human Consultancy Dishonest: 42.00
## Total sampled_consultancies_all_debates_weights_setting for Human Consultancy Honest: 41.00
## Total sampled_consultancies_all_debates_weights_setting for Human Debate: 107.00
```

```
print('Consultancy balanced weights, grouped by Final Setting')
```

```
## Consultancy balanced weights, grouped by Final Setting
```

```
print_weight_summary_by_setting(judgments_online[consultancy_balanced], 'sampled_consultancies_all_deba
```

```
## Total sampled_consultancies_all_debates_weights_grouped_setting for AI Consultancy Dishonest: 38.00
## Total sampled_consultancies_all_debates_weights_grouped_setting for AI Consultancy Honest: 38.00
## Total sampled_consultancies_all_debates_weights_grouped_setting for AI Debate: 75.00
## Total sampled_consultancies_all_debates_weights_grouped_setting for Human Consultancy Dishonest: 30.
## Total sampled_consultancies_all_debates_weights_grouped_setting for Human Consultancy Honest: 30.83
## Total sampled_consultancies_all_debates_weights_grouped_setting for Human Debate: 107.00
```

```
judgments_online = question_weights(
    data=judgments_online,
    columns=['Article ID', 'Question'],
    weight_column_name='sampled_consultancies_debates_weights',
    consultancy_sample=True,
    debate_sample=True
)
judgments_online = question_weights(
    data=judgments_online,
    columns=['Article ID', 'Question', 'Setting'],
    weight_column_name='sampled_consultancies_debates_weights_setting',
    consultancy_sample=True,
    debate_sample=True
```

```
)
judgments_online = question_weights(
    data=judgments_online,
    columns=['Article ID', 'Question', 'Final_Setting'],
    weight_column_name='sampled_consultancies_debates_weights_grouped_setting',
    consultancy_sample=True,
    debate_sample=True
)
```

Note: we are not balancing between settings(?), and more counts of the human debate settings are on the same questions..?

Judge Accuracy Vis

```
import altair
# set graphic parameters
correctColor = "green"
incorrectColor = "crimson"
nullColor = "lightgrey"
onlineColor = "orange"
offlineColor = "blue"
aggColor = "black"
fullWidth = 1400
# fullHeight = 600
def outcomes_by_field(source, rowEncoding = None):
    source['outcome'] = source.apply(
        lambda row: "incomplete" if math.isnan(row['Final probability correct'])
        else "tie" if row['Final probability correct'] == 0.5
        else "correct" if row['Final probability correct'] > 0.5
        else "incorrect",
        axis=1
    )
    source['Final probability correct (with imputation)'] = source.apply(
        lambda row: 0.5 if math.isnan(row['Final probability correct'])
        else row['Final probability correct'],
        axis=1
    )
    source['Final probability correct (dist from half)'] = source.apply(
        lambda row: 0.0 if math.isnan(row['Final probability correct'])
        else abs(row['Final probability correct'] - 0.5),
        axis=1
    )
    if rowEncoding is None:
        groups = ['outcome']
    else:
        groups = ['outcome', rowEncoding.field]
    base = alt.Chart(
        source
    ).transform_joinaggregate(
        groupby=groups,
        group_count='count()'
```

```
    ).encode(
        y=alt.Y('outcome:N', scale=alt.Scale(domain=['correct', 'incorrect', 'tie', 'incomplete']))
    )
    if rowEncoding is not None:
        base = base.encode(row=rowEncoding)
    main_bar = base.mark_bar().encode(
        x=alt.X('count():Q', axis=None),
        color = alt.Color(
            'Final probability correct (with imputation):Q',
            scale=alt.Scale(range=[incorrectColor, nullColor, correctColor], domain=[0.0, 1.0]),
            title='Final Probability\nAssigned to\nCorrect Answer'
        ),
        order=alt.Order(
            'Final probability correct (dist from half):Q',
            sort='ascending'
        ),
        tooltip = [
            'outcome:N',
            alt.Tooltip('group_count:Q', title="Judgments"),
            alt.Tooltip('count():Q', title = 'Judgments with this probability'),
            'Final probability correct:Q'
        ]
    ).properties(width=fullWidth)# height=fullHeight/3)
    return main_bar

def accuracy_by_field(source, by_turn: bool = False, yEncoding = None, invert = False):
    if by_turn:
        prob_correct_field = 'Probability correct'
    else:
        prob_correct_field = 'Final probability correct'
    if source.get('Final probability assigned') is not None:
        prob_assigned_field = 'Final probability assigned'
    else:
        prob_assigned_field = prob_correct_field
    if yEncoding is None:
        groups = []
    else:
        groups = [yEncoding.field]
    base = alt.Chart(source).transform_joinaggregate(
        total = "count()",
        groupby = groups
    ).transform_calculate(
        proportion = '1 / datum.total'
    ).transform_calculate(
        is_correct = f'datum["{prob_correct_field}"] > 0.5 ? 1 : 0',
        is_win = f'datum["{prob_assigned_field}"] > 0.5 ? 1 : 0',
        is_not_correct = f'datum["{prob_correct_field}"] <= 0.5 ? 1 : 0'
    )
    if yEncoding is not None:
        base = base.encode(y=yEncoding)
    main_bar = base.mark_bar().encode(
        x=alt.X('sum(proportion):Q',
            axis=alt.Axis(title=None, format='.0%', labelExpr="(datum.value * 5) % 1 ? null : datum.labe
```

```
            scale=alt.Scale(domain=[0.0, 1.0])
        ),
        color=alt.Color(f'{prob_correct_field}:Q', scale=alt.Scale(range=[incorrectColor, nullColor, co
        order=alt.Order(
            f'{prob_assigned_field}:Q',
            sort='descending' if not invert else 'ascending'
        ),
        tooltip = [
            'count():Q',
            'total:Q',
            'sum(proportion):Q',
            f'{prob_correct_field}:Q',
            'Room name:N',
            'Participant:N'
        ]
    ).properties(width=fullWidth)# height=fullHeight/12)
    prop_color = aggColor
    # rule_thickness = 1.0
    # err_thickness = 1.0
    point_size = 25.0
    mean_field = 'is_win' if not invert else 'is_not_correct'
    gold_err = (base
    ).mark_rule(
        # extent='ci',
        color=prop_color,
    ).encode(
        x=f'ci0({mean_field}):Q',
        x2=f'ci1({mean_field}):Q',
        # scale=alt.Scale(zero=False)
        tooltip=[]
    )
    gold_mean = base.mark_point(
        # thickness=2.0
        color=prop_color, size=point_size, filled=True
    ).encode(
        x=alt.X(f'mean({mean_field}):Q',
            scale=alt.Scale(zero=False)),
    )
    gold_mean_num = base.mark_text(
        color=prop_color,
        align='left',
        baseline='bottom',
        fontSize=24,
        fontWeight='bold',
        dx=4,
        dy=-4
    ).encode(
        text=alt.Text(f'mean({mean_field}):Q', format='.0%'),
        x=alt.X(f'mean({mean_field}):Q',
            scale=alt.Scale(zero=False)),
    )
    return main_bar + gold_err + gold_mean + gold_mean_num
```

```python
def accuracy_by_judge_setting(setting,data_frame_source):
    source = data_frame_source
    yEncoding = alt.Y(field = setting, type='nominal', title=None)
    outcomes_source = source
    accuracy_source = source[source['Final probability correct'].notna()]
    chart = alt.vconcat(
        accuracy_by_field(
            accuracy_source,
            yEncoding = yEncoding
        ).properties(title=alt.TitleParams(text="Judge Accuracy", fontSize=28)),
    ).resolve_scale(x = 'independent')
    return chart.configure(
            padding = {"left": 7, "top": 5, "right": 5, "bottom": 5},
            axis = alt.Axis(labelFontSize=20,labelLimit=300),
            legend = alt.LegendConfig(disable = True)
            ).configure_view(
        step=65,  # adjust the step parameter for margins
        )

accuracy_by_judge_setting(setting = 'Final_Setting', data_frame_source = judgments_online.loc[
    (judgments_online['Consultancy Sample'] == True) |
    (~judgments_online['Final_Setting'].str.contains("Consultancy", na=False))
])
```

```python
#chart.save('judge_accuracy_settings.png', scale_factor=4)
```

```python
consultancies = judgments_online.loc[judgments_online['Consultancy Sample'] == True]
consultancies['Setting'] = consultancies['Setting'].apply(lambda x: ' '.join(x.split()[:-1]) + f" ({x.sp
accuracy_by_judge_setting(setting = 'Setting', data_frame_source = consultancies)
```

```python
sample = judgments_online.loc[
    (judgments_online['Consultancy Sample'] == True) |
    (~judgments_online['Final_Setting'].str.contains("Consultancy", na=False))
]
```

Load into R environment

```r
sample <- py$sample
sample <- sample[,c("Room name", "Participant")]
write.csv(sample, "/Users/bila/Downloads/python_sample.csv")
set.seed(123)
# Read in objects from Python with py$
judgments <- py$judgments
judgments_online <- py$judgments_online
correctColor = "#008000"
incorrectColor = "#DC143C"
# Change type into factor so it is read as categories which can be manipulated instead of characters
judgments_online$Participant <- as.factor(judgments_online$Participant)
judgments_online$Setting <- as.factor(judgments_online$Setting)
```

```r
# Doing some sanity checks
subset_dishonest <- judgments_online[judgments_online$`Human Consultancy Sample` == TRUE & judgments_onl
subset_honest <- judgments_online[judgments_online$`Human Consultancy Sample` == TRUE & judgments_onlin
#Are the question weights equal for human consultancies?"
table(subset_dishonest$sampled_consultancies_all_debates_weights_grouped_setting) ; table(subset_honest$
```

```
##
##              0.25 0.333333333333333               0.5                 1
##                 2                 5                26                15

##
##              0.25 0.333333333333333               0.5                 1
##                 2                10                18                18
```

```r
#What does the accuracy look like for those question weights?
#table(subset_dishonest$sampled_consultancies_all_debates_weights_grouped_setting, subset_dishonest$Fin
#table(subset_honest$sampled_consultancies_all_debates_weights_grouped_setting, subset_honest$Final_Acc



#subset_human_consultancies <- judgments_online[judgments_online$`Human Consultancy Sample` == TRUE & j
#table(subset_human_consultancies$sampled_consultancies_all_debates_weights_grouped_setting, subset_hum
#Difference between grouping and not grouping question weights
table(judgments_online$Final_Setting, judgments_online$sampled_consultancies_all_debates_weights_groupe
```

```
##
##                      0 0.25 0.333333333333333 0.5  1
##   AI Consultancy    17    0                 0   0 76
##   AI Debate          0    0                 0  24 63
##   Human Consultancy 11    4                15  44 33
##   Human Debate       0    0                 0  94 60

##
##                      0 0.166666666666667 0.2 0.25 0.333333333333333 0.5  1
##   AI Consultancy    17                 2   8    4                 1   0 61
##   AI Debate          0                 4  14    6                 0   3 60
##   Human Consultancy 11                 3  19   24                26  18  6
##   Human Debate       0                 3  14   14                27  61 35
```

```r
# Balanced consultancies difference between grouping and not grouping question weights
consultancy_condition <- (judgments_online$Sample == TRUE) | (!grepl("Consultancy", judgments_online$Fi
table(judgments_online[consultancy_condition, ]$Final_Setting, judgments_online[consultancy_condition, 
```

```
## , ,  = FALSE
##
##
##                     0.25 0.333333333333333 0.5  1
##   AI Consultancy       0                 0   0 13
##   AI Debate            0                 0   7 12
##   Human Consultancy    3                 5  14  5
##   Human Debate         0                 0  16  8
```

```
##
## , ,  = TRUE
##
##
##                       0.25 0.333333333333333 0.5  1
##   AI Consultancy       0                 0   0 63
##   AI Debate            0                 0  17 51
##   Human Consultancy    1                10  30 28
##   Human Debate         0                 0  78 52
```

```
table(judgments_online[consultancy_condition, ]$Final_Setting, judgments_online[consultancy_condition, ]
```

```
## , ,  = FALSE
##
##
##                       0.166666666666667 0.2 0.25 0.333333333333333 0.5  1
##   AI Consultancy                      0   1    0                 0   0 12
##   AI Debate                           1   3    1                 0   2 12
##   Human Consultancy                   1   5    9                 7   3  2
##   Human Debate                        0   5    2                 5   9  3
##
## , ,  = TRUE
##
##
##                       0.166666666666667 0.2 0.25 0.333333333333333 0.5  1
##   AI Consultancy                      2   7    4                 1   0 49
##   AI Debate                           3  11    5                 0   1 48
##   Human Consultancy                   2  14   15                19  15  4
##   Human Debate                        3   9   12                22  52 32
```

```
# Sampled data (balanced consultancies and sampled debates) difference between grouping and not groupin
table(judgments_online[judgments_online$Sample == TRUE, ]$Final_Setting, judgments_online[judgments_onli
```

```
##
##                       0.25 0.333333333333333 0.5   1
##   AI Consultancy       0                 0   0  76
##   AI Debate            0                 0   0  75
##   Human Consultancy    4                15  44  33
##   Human Debate         0                 0   0 107
```

```
table(judgments_online[judgments_online$Sample == TRUE, ]$Final_Setting, judgments_online[judgments_onli
```

```
##
##                       0.2 0.25 0.333333333333333 0.5  1
##   AI Consultancy        1   11                 3   0 61
##   AI Debate             1   10                 2   1 61
##   Human Consultancy     2   32                28  28  6
##   Human Debate          1   15                12  17 62
```

Robustness Checks

```r
# read other sampling
sample.rooms <- read.csv("~/Downloads/sample-rooms-2.csv", header=FALSE)
# Check whether chosen sample in sample.rooms is the same as judgments_online
# based on columns V2 and V1 in sample.rooms and Participant and `Room name` in judgments_online
sample.rooms_samples <- sort(paste0(sample.rooms$V2, sample.rooms$V1))
judgments_online_samples <- paste0(judgments_online[consultancy_condition,]$Participant, judgments_onli

missing_sample.room <- sample.rooms[sample.rooms_samples %in% judgments_online_samples == FALSE, ]
sampled_judgments_online <- judgments_online[consultancy_condition,]
missing_judgments_online <- sampled_judgments_online[judgments_online_samples %in% sample.rooms_samples

judgments_online$check <- paste0(judgments_online$Participant, judgments_online$`Room name`)
matching_sampled_judgments_online <- subset(judgments_online, judgments_online$check %in% sample.rooms_
rooms_hc <- subset(matching_sampled_judgments_online, matching_sampled_judgments_online$Final_Setting ==
```

```python
different_sample = r.rooms_hc
different_sample.groupby(['Question', 'Article ID']).size().value_counts().sum()
```

```
## 61
```

```python
judgments_online[(judgments_online['Setting'].str.contains('Human Consultancy')) & (judgments_online['C
```

```
## 61
```

```python
filtered_df1 = different_sample.groupby(['Question', 'Article ID']).filter(lambda x: len(x) > 2)
filtered_df2 = different_sample.groupby(['Question', 'Article ID']).filter(lambda x: len(x) <= 2)

filtered_df1["Untimed annotator context bins"].value_counts()
```

```
## 3    13
## 2     7
## 4     0
## 1     0
## Name: Untimed annotator context bins, dtype: int64
```

```python
filtered_df2["Untimed annotator context bins"].value_counts()
```

```
## 2    30
## 3    28
## 4    18
## 1     0
## Name: Untimed annotator context bins, dtype: int64
```

```python
filtered_df1["Final_Accuracy"].mean()
```

```
## 0.65
```

```
filtered_df2["Final_Accuracy"].mean()
```

```
## 0.7631578947368421
```

```
judgments_online[judgments_online['Final_Setting']=="Human Debate"].groupby(['Question', 'Article ID'])
```

```
## 1    60
## 2    47
## dtype: int64
```

```
judgments_online[judgments_online['Final_Setting']=="AI Debate"].groupby(['Question', 'Article ID']).si
```

```
## 1    63
## 2    12
## dtype: int64
```

```
paste("Overall variance is",
      var(judgments_online$Final_Accuracy), "(mean way)",
      ((sum(judgments_online$Final_Accuracy, na.rm = T) / length(judgments_online$Final_Accuracy)) * (1
```

```
## [1] "Overall variance is 0.166790352504638 (mean way) 0.000378209416110291 (prop way)"
```

```
# Accuracy variation per setting
judgments_online %>%
  group_by(Final_Setting) %>%
  summarise(
    var_mean = var(Final_Accuracy),
    n = length(Final_Accuracy),
    x_aka_num_correct = sum(Final_Accuracy),
    p_aka_accuracy = (x_aka_num_correct / n),
    var_prop = (p_aka_accuracy * (1 - p_aka_accuracy)) / (n - 1)
  ) %>% mutate(avg_var_mean = mean(var_mean, na.rm = T),
               avg_var_prop = mean(var_prop, na.rm = T))
```

```
## # A tibble: 4 x 8
##   Final_Setting     var_mean     n x_aka_num_correct p_aka_accuracy var_prop
##   <chr>                <dbl> <int>             <int>          <dbl>    <dbl>
## 1 AI Consultancy       0.158    93                75          0.806 0.00170
## 2 AI Debate            0.173    87                68          0.782 0.00198
## 3 Human Consultancy    0.212   107                75          0.701 0.00198
## 4 Human Debate         0.132   154               130          0.844 0.000860
## # i 2 more variables: avg_var_mean <dbl>, avg_var_prop <dbl>
```

```
# Accuracy variation per setting (consultancies balanced)
judgments_online[consultancy_condition, ] %>%
  group_by(Final_Setting) %>%
  summarise(
    var_mean = var(Final_Accuracy),
    n = length(Final_Accuracy),
```

```r
    x_aka_num_correct = sum(Final_Accuracy),
    p_aka_accuracy = (x_aka_num_correct / n),
    var_prop = (p_aka_accuracy * (1 - p_aka_accuracy)) / (n - 1)
  ) %>% mutate(avg_var_mean = mean(var_mean, na.rm = T),
               avg_var_prop = mean(var_prop, na.rm = T))
```

```
## # A tibble: 4 x 8
##   Final_Setting     var_mean     n x_aka_num_correct p_aka_accuracy var_prop
##   <chr>                <dbl> <int>             <int>          <dbl>    <dbl>
## 1 AI Consultancy       0.144    76                63          0.829 0.00189
## 2 AI Debate            0.173    87                68          0.782 0.00198
## 3 Human Consultancy    0.204    96                69          0.719 0.00213
## 4 Human Debate         0.132   154               130          0.844 0.000860
## # i 2 more variables: avg_var_mean <dbl>, avg_var_prop <dbl>
```

```r
judgments_online %>%
  group_by(base_room_name) %>%
  summarise(
    var_mean = var(Final_Accuracy),
    n = length(Final_Accuracy),
    x_aka_num_correct = sum(Final_Accuracy),
    p_aka_accuracy = (x_aka_num_correct / n),
    var_prop = (p_aka_accuracy * (1 - p_aka_accuracy)) / (n - 1)
  ) %>% mutate(avg_var_mean = mean(var_mean, na.rm = T),
               avg_var_prop = mean(var_prop, na.rm = T))
```

```
## # A tibble: 100 x 8
##    base_room_name       var_mean     n x_aka_num_correct p_aka_accuracy var_prop
##    <chr>                   <dbl> <int>             <int>          <dbl>    <dbl>
##  1 a-pail-of-air-          0.25      4                 3          0.75    0.0625
##  2 a-planet-named-joe-     0         6                 6          1       0
##  3 ambition-               0.25      4                 3          0.75    0.0625
##  4 atom-mystery-young-~    0.267     6                 4          0.667   0.0444
##  5 break-a-leg-            0.25      4                 3          0.75    0.0625
##  6 cakewalk-to-gloryan~    0.5       2                 1          0.5     0.25
##  7 call-him-nemesis-       0.267     6                 4          0.667   0.0444
##  8 captain-chaos-          0.2       5                 4          0.8     0.04
##  9 castaways-of-eros-      0.2       5                 4          0.8     0.04
## 10 coming-of-the-gods-     0.2       5                 4          0.8     0.04
## # i 90 more rows
## # i 2 more variables: avg_var_mean <dbl>, avg_var_prop <dbl>
```

```r
judgments_online %>%
  group_by(Question) %>%
  summarise(
    var_mean = var(Final_Accuracy),
    n = length(Final_Accuracy),
    x_aka_num_correct = sum(Final_Accuracy),
    p_aka_accuracy = (x_aka_num_correct / n),
    var_prop = (p_aka_accuracy * (1 - p_aka_accuracy)) / (n - 1)
  ) %>% mutate(avg_var_mean = mean(var_mean, na.rm = T),
               avg_var_prop = mean(var_prop, na.rm = T))
```

```
## # A tibble: 252 x 8
##    Question              var_mean     n x_aka_num_correct p_aka_accuracy var_prop
##    <chr>                    <dbl> <int>             <int>          <dbl>    <dbl>
##  1 "According to Retie~   NA        1                 1          1      NaN
##  2 "After a short time~   NA        1                 0          0      NaN
##  3 "After reading abou~   NA        1                 0          0      NaN
##  4 "Between Martians a~   NA        1                 1          1      NaN
##  5 "By the end of the ~   NA        1                 0          0      NaN
##  6 "By the end of the ~    0.267    6                 4          0.667    0.0444
##  7 "Did the characters~   NA        1                 1          1      NaN
##  8 "Did the questions ~    0.167    6                 5          0.833    0.0278
##  9 "From the informati~    0.2      5                 4          0.8      0.04
## 10 "Generally, which o~    0        2                 2          1        0
## # i 242 more rows
## # i 2 more variables: avg_var_mean <dbl>, avg_var_prop <dbl>
```

```r
judgments_online[consultancy_condition, ] %>%
  group_by(base_room_name) %>%
  summarise(
    var_mean = var(Final_Accuracy),
    n = length(Final_Accuracy),
    x_aka_num_correct = sum(Final_Accuracy),
    p_aka_accuracy = (x_aka_num_correct / n),
    var_prop = (p_aka_accuracy * (1 - p_aka_accuracy)) / (n - 1)
  ) %>% mutate(avg_var_mean = mean(var_mean, na.rm = T),
               avg_var_prop = mean(var_prop, na.rm = T))
```

```
## # A tibble: 100 x 8
##    base_room_name       var_mean     n x_aka_num_correct p_aka_accuracy var_prop
##    <chr>                   <dbl> <int>             <int>          <dbl>    <dbl>
##  1 a-pail-of-air-           0.25     4                 3           0.75   0.0625
##  2 a-planet-named-joe-      0        5                 5           1      0
##  3 ambition-                0.25     4                 3           0.75   0.0625
##  4 atom-mystery-young-~     0.2      5                 4           0.8    0.04
##  5 break-a-leg-             0.25     4                 3           0.75   0.0625
##  6 cakewalk-to-gloryan~     0.5      2                 1           0.5    0.25
##  7 call-him-nemesis-        0.2      5                 4           0.8    0.04
##  8 captain-chaos-           0.2      5                 4           0.8    0.04
##  9 castaways-of-eros-       0.25     4                 3           0.75   0.0625
## 10 coming-of-the-gods-      0.25     4                 3           0.75   0.0625
## # i 90 more rows
## # i 2 more variables: avg_var_mean <dbl>, avg_var_prop <dbl>
```

```r
judgments_online[consultancy_condition, ] %>%
  group_by(Question) %>%
  summarise(
    var_mean = var(Final_Accuracy),
    n = length(Final_Accuracy),
    x_aka_num_correct = sum(Final_Accuracy),
    p_aka_accuracy = (x_aka_num_correct / n),
    var_prop = (p_aka_accuracy * (1 - p_aka_accuracy)) / (n - 1)
  ) %>% mutate(avg_var_mean = mean(var_mean, na.rm = T),
               avg_var_prop = mean(var_prop, na.rm = T))
```

```
## # A tibble: 246 x 8
##    Question              var_mean     n x_aka_num_correct p_aka_accuracy var_prop
##    <chr>                    <dbl> <int>             <int>          <dbl>    <dbl>
##  1 "According to Retie~      NA       1                 1            1      NaN
##  2 "After a short time~      NA       1                 0            0      NaN
##  3 "After reading abou~      NA       1                 0            0      NaN
##  4 "Between Martians a~      NA       1                 1            1      NaN
##  5 "By the end of the ~      NA       1                 0            0      NaN
##  6 "By the end of the ~       0.3     5                 3            0.6      0.06
##  7 "Did the characters~      NA       1                 1            1      NaN
##  8 "Did the questions ~       0       5                 5            1        0
##  9 "From the informati~       0.2     5                 4            0.8      0.04
## 10 "Generally, which o~       0       2                 2            1        0
## # i 236 more rows
## # i 2 more variables: avg_var_mean <dbl>, avg_var_prop <dbl>
```

```
judgments_online[consultancy_condition,] %>%
  group_by(base_room_name) %>%
  summarise(
    var_mean = var(Final_Accuracy, na.rm = T),
    n = length(Final_Accuracy),
    x_aka_num_correct = sum(Final_Accuracy, na.rm = T),
    p_aka_accuracy = (x_aka_num_correct / n),
    var_prop = (p_aka_accuracy * (1 - p_aka_accuracy)) / (n - 1)
  ) %>% summarise(avg_var_mean = mean(var_mean, na.rm = T),
              avg_var_prop = mean(var_prop, na.rm = T))
```

```
## # A tibble: 1 x 2
##   avg_var_mean avg_var_prop
##          <dbl>        <dbl>
## 1        0.164       0.0423
```

```
judgments_online[consultancy_condition,] %>%
  group_by(Question) %>%
  summarise(
    var_mean = var(Final_Accuracy, na.rm = T),
    n = length(Final_Accuracy),
    x_aka_num_correct = sum(Final_Accuracy, na.rm = T),
    p_aka_accuracy = (x_aka_num_correct / n),
    var_prop = (p_aka_accuracy * (1 - p_aka_accuracy)) / (n - 1)
  ) %>% summarise(avg_var_mean = mean(var_mean, na.rm = T),
              avg_var_prop = mean(var_prop, na.rm = T))
```

```
## # A tibble: 1 x 2
##   avg_var_mean avg_var_prop
##          <dbl>        <dbl>
## 1        0.159       0.0588
```

Results

Difference in Accuracy

```r
# Make a function to easily try out different weights
acc_diff_test <- function(design, Setting){
  print(design)
  freq_table <- svytable(~Final_Setting+Final_Accuracy, design)
  chisq_result <- svychisq(~Final_Setting+Final_Accuracy, design, statistic = "Chisq")
  print(chisq_result)
  pairwise_result <- pairwise.prop.test(freq_table, p.adjust.method="none", alternative="two.sided")
  print(pairwise_result)
  freq_table <- cbind(freq_table, Accuracy = (freq_table[,2] / (freq_table[,1]+freq_table[,2]))*100)
  print(freq_table)
}

print("Really raw")
```

```
## [1] "Really raw"
```

```r
acc_diff_test(svydesign(ids = ~1, data = judgments))
```

```
## Warning in svydesign.default(ids = ~1, data = judgments): No weights or
## probabilities supplied, assuming equal probability
```

```
## Independent Sampling design (with replacement)
## print(design)
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  svychisq(~Final_Setting + Final_Accuracy, design, statistic = "Chisq")
## X-squared = 17.998, df = 3, p-value = 0.0004457
##
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  freq_table
##
##                   AI Consultancy AI Debate Human Consultancy
## AI Debate         0.881          -         -
## Human Consultancy 0.148          0.277     -
## Human Debate      0.129          0.052     0.000056
##
## P value adjustment method: none
##                   FALSE TRUE Accuracy
## AI Consultancy       19   77 80.20833
## AI Debate            20   72 78.26087
## Human Consultancy    36   87 70.73171
## Human Debate         50  333 86.94517
```

```r
print("Raw")
```

```
## [1] "Raw"
```

```r
acc_diff_test(svydesign(ids = ~1, data = judgments_online))
```

```
## Warning in svydesign.default(ids = ~1, data = judgments_online): No weights or
## probabilities supplied, assuming equal probability

## Independent Sampling design (with replacement)
## print(design)
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  svychisq(~Final_Setting + Final_Accuracy, design, statistic = "Chisq")
## X-squared = 8.0006, df = 3, p-value = 0.04637
##
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  freq_table
##
##                   AI Consultancy AI Debate Human Consultancy
## AI Debate         0.8200         -         -
## Human Consultancy 0.1199         0.2689    -
## Human Debate      0.5555         0.2970    0.0088
##
## P value adjustment method: none
##                   FALSE TRUE Accuracy
## AI Consultancy       18   75 80.64516
## AI Debate            19   68 78.16092
## Human Consultancy    32   75 70.09346
## Human Debate         24  130 84.41558
```

```r
print("Balanced consultancies, NO weights") # still sig
```

```
## [1] "Balanced consultancies, NO weights"
```

```r
acc_diff_test(svydesign(ids = ~1, data = subset(judgments_online, `Consultancy Sample` == TRUE | !grepl
```

```
## Warning in svydesign.default(ids = ~1, data = subset(judgments_online,
## 'Consultancy Sample' == : No weights or probabilities supplied, assuming equal
## probability

## Independent Sampling design (with replacement)
## print(design)
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  svychisq(~Final_Setting + Final_Accuracy, design, statistic = "Chisq")
```

```
## X-squared = 6.3939, df = 3, p-value = 0.09458
##
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  freq_table
##
##                   AI Consultancy AI Debate Human Consultancy
## AI Debate         0.575          -         -
## Human Consultancy 0.129          0.419     -
## Human Debate      0.917          0.297     0.026
##
## P value adjustment method: none
##                   FALSE TRUE Accuracy
## AI Consultancy       13   63 82.89474
## AI Debate            19   68 78.16092
## Human Consultancy    27   69 71.87500
## Human Debate         24  130 84.41558
```

```
print("Balanced consultancies, question weights (grouped settings)")
```

```
## [1] "Balanced consultancies, question weights (grouped settings)"
```

```
acc_diff_test(svydesign(ids = ~1, data = subset(judgments_online, `Consultancy Sample` == TRUE | !grepl
```

```
## Independent Sampling design (with replacement)
## print(design)
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  svychisq(~Final_Setting + Final_Accuracy, design, statistic = "Chisq")
## X-squared = 2.9692, df = 3, p-value = 0.4323
##
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  freq_table
##
##                   AI Consultancy AI Debate Human Consultancy
## AI Debate         0.73           -         -
## Human Consultancy 0.46           0.84      -
## Human Debate      0.85           0.42      0.23
##
## P value adjustment method: none
##                      FALSE     TRUE Accuracy
## AI Consultancy    13.00000 63.00000 82.89474
## AI Debate         15.50000 59.50000 79.33333
## Human Consultancy 14.41667 46.58333 76.36612
## Human Debate      16.00000 91.00000 85.04673
```

```
print("Balanced # consultancies, question weights")
```

```
## [1] "Balanced # consultancies, question weights"
```

```r
acc_diff_test(svydesign(ids = ~1, data = subset(judgments_online, `Consultancy Sample` == TRUE | !grepl
```

```
## Independent Sampling design (with replacement)
## print(design)
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  svychisq(~Final_Setting + Final_Accuracy, design, statistic = "Chisq")
## X-squared = 5.9366, df = 3, p-value = 0.1828
##
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  freq_table
##
##                   AI Consultancy AI Debate Human Consultancy
## AI Debate         0.93           -         -
## Human Consultancy 0.49           0.66      -
## Human Debate      0.46           0.28      0.12
##
## P value adjustment method: none
##                      FALSE     TRUE Accuracy
## AI Consultancy    12.20000 52.06667 81.01660
## AI Debate         14.01667 52.45000 78.91174
## Human Consultancy  9.25000 24.71667 72.76742
## Human Debate      10.66667 70.63333 86.87987
```

```r
print("Balanced consultancies sampled debates, NO weights")
```

```
## [1] "Balanced consultancies sampled debates, NO weights"
```

```r
acc_diff_test(svydesign(ids = ~1, data = subset(judgments_online, `Sample` == TRUE)))
```

```
## Warning in svydesign.default(ids = ~1, data = subset(judgments_online, Sample
## == : No weights or probabilities supplied, assuming equal probability
```

```
## Independent Sampling design (with replacement)
## print(design)
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  svychisq(~Final_Setting + Final_Accuracy, design, statistic = "Chisq")
## X-squared = 6.798, df = 3, p-value = 0.07929
##
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  freq_table
##
##                   AI Consultancy AI Debate Human Consultancy
## AI Debate         0.804          -         -
```

```
## Human Consultancy 0.129          0.296     -
## Human Debate      0.716          0.386     0.021
##
## P value adjustment method: none
##                    FALSE TRUE Accuracy
## AI Consultancy       13   63 82.89474
## AI Debate            15   60 80.00000
## Human Consultancy    27   69 71.87500
## Human Debate         15   92 85.98131
```

```
print("Balanced consultancies sampled debates, question weights (grouped settings)")
```

```
## [1] "Balanced consultancies sampled debates, question weights (grouped settings)"
```

```
acc_diff_test(svydesign(ids = ~1, data = subset(judgments_online, `Sample` == TRUE), weights = ~sampled_
```

```
## Independent Sampling design (with replacement)
## print(design)
##
##  Pearson's X^2: Rao & Scott adjustment
##
## data:  svychisq(~Final_Setting + Final_Accuracy, design, statistic = "Chisq")
## X-squared = 3.0023, df = 3, p-value = 0.4009
##
##
##  Pairwise comparisons using Pairwise comparison of proportions
##
## data:  freq_table
##
##                   AI Consultancy AI Debate Human Consultancy
## AI Debate         0.80           -         -
## Human Consultancy 0.46           0.76      -
## Human Debate      0.72           0.39      0.17
##
## P value adjustment method: none
##                      FALSE     TRUE Accuracy
## AI Consultancy    13.00000 63.00000 82.89474
## AI Debate         15.00000 60.00000 80.00000
## Human Consultancy 14.41667 46.58333 76.36612
## Human Debate      15.00000 92.00000 85.98131
```

```
svytable(~Final_Setting+Final_Accuracy, svydesign(ids = ~1, data = subset(judgments_online, `Sample` ==
```

```
## Warning in svydesign.default(ids = ~1, data = subset(judgments_online, Sample
## == : No weights or probabilities supplied, assuming equal probability
```

```
##                    Final_Accuracy
## Final_Setting       FALSE TRUE
##   AI Consultancy       13   63
##   AI Debate            15   60
##   Human Consultancy    27   69
##   Human Debate         15   92
```

```r
svytable(~Final_Setting+Final_Accuracy, svydesign(ids = ~1, data = subset(judgments_online, `Sample` ==
```

```
##                    Final_Accuracy
## Final_Setting           FALSE     TRUE
##   AI Consultancy    13.00000 63.00000
##   AI Debate         15.00000 60.00000
##   Human Consultancy 14.41667 46.58333
##   Human Debate      15.00000 92.00000
```

```r
print("Now trying manually tests that aren't pairwise + cobfidence intervals for the table")
```

```
## [1] "Now trying manually tests that aren't pairwise + cobfidence intervals for the table"
```

```r
process_table <- function(svy_table, round_by) {
  # Ensure that the input is a svytable object
  if (!inherits(svy_table, "svytable")) {
    stop("Input must be a svytable object")
  }
  # Add accuracy
  svy_table <- cbind(svy_table, Accuracy = (svy_table[,2] / (svy_table[,1] + svy_table[,2])) * 100)
  # Calculate the difference in accuracy for each row compared to "Human Debate"
  difference_with_debate <- svy_table[,"Accuracy"] - svy_table["Human Debate", "Accuracy"]
  # Bind the difference column to the svy_table
  svy_table <- cbind(svy_table, `Difference with Debate` = difference_with_debate)
  # Initialize vectors to store confidence interval bounds and p-values
  ci_lowers <- c() ; ci_uppers <- c() ; p_values <- c()
  # Loop through each setting
  for (setting in rownames(svy_table)) {
    # Use prop.test to compare the setting's accuracy with "Human Debate"
    results <- prop.test(
      x = c(svy_table[setting, "TRUE"], svy_table["Human Debate", "TRUE"]),
      n = c((svy_table[setting, "TRUE"] + svy_table[setting, "FALSE"]), (svy_table["Human Debate", "TRUE
      correct = F
    )
    # Extract the confidence interval and store it as a string in the format "lower - upper"
    ci_lower <- round(results$conf.int[1] * 100,round_by)  # Multiply by 100 to convert to percentage
    ci_upper <- round(results$conf.int[2] * 100,round_by)  # Multiply by 100 to convert to percentage
    ci_lowers <- c(ci_lowers, ci_lower)
    ci_uppers <- c(ci_uppers, ci_upper)
    p_values <- c(p_values, results$p.value)
  }
  # Change to wanted format (judgments summed, split counts removed)
  svy_table <- cbind("n Judgments" = (svy_table[,"FALSE"] + svy_table[,"TRUE"]), svy_table)
  svy_table <- svy_table[ , !(colnames(svy_table) %in% c("FALSE", "TRUE"))]
  # Concatenate the CI bounds into a single string
  ci_strings <- paste0("[", ci_lowers, ", ", ci_uppers, "]")
  # Convert svy_table to a data.frame so adding the strings doesn't change the data type for entire mat
  svy_table <- as.data.frame(svy_table)
  # Bind the confidence interval bounds and p-values to the svy_table
  svy_table <- cbind(svy_table, `95% CI [lower, upper]` = ci_strings, `p val` = p_values)
  return(svy_table)
}
```

```r
# First table, all data accuracy
svy_table_input <- svytable(
  ~Final_Setting + Final_Accuracy,
  design = svydesign(
    ids = ~1,
    data = subset(judgments_online, `Consultancy Sample` == TRUE | !grepl("Consultancy", Final_Setting))
  )
)
```

```
## Warning in svydesign.default(ids = ~1, data = subset(judgments_online,
## ‘Consultancy Sample‘ == : No weights or probabilities supplied, assuming equal
## probability
```

```r
svy_table_input_2 <- svytable(
  ~Final_Setting + Final_Accuracy,
  design = svydesign(
    ids = ~1,
    data = matching_sampled_judgments_online,
  )
)
```

```
## Warning in svydesign.default(ids = ~1, data =
## matching_sampled_judgments_online, : No weights or probabilities supplied,
## assuming equal probability
```

```r
# Call the function
final_table <- process_table(svy_table_input, round_by = 3)
final_table
```

```
##                    n Judgments Accuracy Difference with Debate
## AI Consultancy              76 82.89474              -1.520848
## AI Debate                   87 78.16092              -6.254665
## Human Consultancy           96 71.87500             -12.540584
## Human Debate               154 84.41558               0.000000
##                    95% CI [lower, upper]      p val
## AI Consultancy          [-11.743, 8.701] 0.76777832
## AI Debate               [-16.656, 4.147] 0.22320432
## Human Consultancy      [-23.204, -1.877] 0.01670386
## Human Debate             [-8.101, 8.101] 1.00000000
```

```r
final_table_2 <- process_table(svy_table_input_2, round_by = 3)
final_table_2
```

```
##                    n Judgments Accuracy Difference with Debate
## AI Consultancy              76 80.26316              -4.152427
## AI Debate                   87 78.16092              -6.254665
## Human Consultancy           96 73.95833             -10.457251
## Human Debate               154 84.41558               0.000000
##                    95% CI [lower, upper]      p val
## AI Consultancy          [-14.777, 6.472] 0.42989215
## AI Debate               [-16.656, 4.147] 0.22320432
## Human Consultancy        [-20.94, 0.025] 0.04278964
## Human Debate             [-8.101, 8.101] 1.00000000
```

```r
knitr::kable(final_table, booktab = TRUE, digits = c(rep(3,3),NA,3))
```

	n Judgments	Accuracy	Difference with Debate	95% CI [lower, upper]	p val
AI Consultancy	76	82.895	-1.521	[-11.743, 8.701]	0.768
AI Debate	87	78.161	-6.255	[-16.656, 4.147]	0.223
Human Consultancy	96	71.875	-12.541	[-23.204, -1.877]	0.017
Human Debate	154	84.416	0.000	[-8.101, 8.101]	1.000

```r
knitr::kable(final_table_2, booktab = TRUE, digits = c(rep(3,3),NA,3))
```

	n Judgments	Accuracy	Difference with Debate	95% CI [lower, upper]	p val
AI Consultancy	76	80.263	-4.152	[-14.777, 6.472]	0.430
AI Debate	87	78.161	-6.255	[-16.656, 4.147]	0.223
Human Consultancy	96	73.958	-10.457	[-20.94, 0.025]	0.043
Human Debate	154	84.416	0.000	[-8.101, 8.101]	1.000

```r
svy_table <- svytable(
  ~Final_Setting + Final_Accuracy,
  design = svydesign(
    ids = ~1,
    data = subset(judgments_online, `Consultancy Sample` == TRUE | !grepl("Consultancy", Final_Setting)
  )
)
```

```
## Warning in svydesign.default(ids = ~1, data = subset(judgments_online,
## 'Consultancy Sample' == : No weights or probabilities supplied, assuming equal
## probability
```

```r
prop.test(
      x = c(svy_table["Human Consultancy", "TRUE"], svy_table["Human Debate", "TRUE"]),
      n = c((svy_table["Human Consultancy", "TRUE"] + svy_table["Human Consultancy", "FALSE"]), (svy_ta
    )
```

```
##
##  2-sample test for equality of proportions with continuity correction
##
## data:  c(svy_table["Human Consultancy", "TRUE"], svy_table["Human Debate", "TRUE"]) out of c((svy_ta
## X-squared = 4.981, df = 1, p-value = 0.02563
## alternative hypothesis: two.sided
## 95 percent confidence interval:
##  -0.24049410 -0.01031759
## sample estimates:
##    prop 1    prop 2
## 0.7187500 0.8441558
```

```r
# # Possible table?, high confidence accuracy
# high_conf_data <- subset(judgments_online,
#                           `Final probability correct` <= 0.01 | `Final probability correct` >= 0.99)

# # Create the svytable object for high confidence accuracy
```

```
# svy_table_high_conf <- svytable(
#   ~Final_Setting + Final_Accuracy,
#   design = svydesign(
#     ids = ~1,
#     data = subset(high_conf_data, `Consultancy Sample` == TRUE | !grepl("Consultancy", Final_Setting)),
#     weights = ~sampled_consultancies_all_debates_weights_grouped_setting
#   )
# )

# # Call the function for high confidence accuracy
# high_conf_table <- process_table(svy_table_high_conf, round_by = 1)
# high_conf_table

# # Render the high confidence accuracy table
# knitr::kable(high_conf_table, booktab = TRUE, digits = c(rep(1,3),NA,3))
```

```
# # Possible table?, high confidence accuracy
# low_conf_data <- subset(judgments_online,
#                          `Final probability correct` >= 0.30 & `Final probability correct` <= 0.70)

# # Create the svytable object for high confidence accuracy
# svy_table_low_conf <- svytable(
#   ~Final_Setting + Final_Accuracy,
#   design = svydesign(
#     ids = ~1,
#     data = subset(low_conf_data, `Consultancy Sample` == TRUE | !grepl("Consultancy", Final_Setting))
#     weights = ~sampled_consultancies_all_debates_weights_grouped_setting
#   )
# )

# Call the function for high confidence accuracy
#low_conf_table <- process_table(svy_table_low_conf, round_by = 1)
#low_conf_table

# Render the high confidence accuracy table
#knitr::kable(low_conf_table, booktab = TRUE, digits = c(rep(1,3),NA,3))
```

Difference in final probability correct

```
judgments_online$`Reward penalty 0.5` <- log2(judgments_online$`Final probability correct`) - 0.5*(judg
judgments_online$fpc <- judgments_online$`Final probability correct`

# Weighted Kruskal-Wallis
svyranktest(fpc~Final_Setting, svydesign(ids = ~1, data = subset(judgments_online, `Consultancy Sample`
```

```
##
```

```
##  Design-based KruskalWallis test
##
## data:  fpc ~ Final_Setting
## df = 3, Chisq = 7.3067, p-value = 0.06431
```

```
# Test Human Settings only
svyranktest(fpc~Final_Setting,
            svydesign(ids = ~1, data = subset(judgments_online, `Human Consultancy Sample` == TRUE | !g
            test = "wilcoxon")
```

```
##
##  Design-based KruskalWallis test
##
## data:  fpc ~ Final_Setting
## t = 1.5183, df = 248, p-value = 0.1302
## alternative hypothesis: true difference in mean rank score is not equal to 0
## sample estimates:
## difference in mean rank score
##                     0.0594665
```

```
svyranktest(fpc~Final_Setting,
            svydesign(ids = ~1, data = subset(judgments_online, `Human Consultancy Sample` == TRUE | !g
            test = "median")
```

```
##
##  Design-based median test
##
## data:  fpc ~ Final_Setting
## t = 1.5865, df = 248, p-value = 0.1139
## alternative hypothesis: true difference in mean rank score is not equal to 0
## sample estimates:
## difference in mean rank score
##                     0.1117282
```

```
# TODO: check test for human consultancy & human debate, make table. Might have to rebuild package to g
# Note: see publication in help page for more

# all
pairwise.wilcox.test(judgments_online$`Final probability correct`, judgments_online$Final_Setting)
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  judgments_online$`Final probability correct` and judgments_online$Final_Setting
##
##                   AI Consultancy AI Debate Human Consultancy
## AI Debate         1.0000         -         -
## Human Consultancy 0.0217         0.0153    -
## Human Debate      1.0000         1.0000    0.0063
##
## P value adjustment method: holm
```

```r
# human settings
filtered_data <- judgments_online[judgments_online$Final_Setting %in% c("Human Consultancy", "Human Deba
wilcox.test(
  `Final probability correct` ~ Final_Setting,
  data = filtered_data,
  paired = FALSE,
  conf.int = TRUE
)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  Final probability correct by Final_Setting
## W = 6308.5, p-value = 0.00105
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.0900006180 -0.0000116136
## sample estimates:
## difference in location
##            -0.04993806
```

```r
wilcox.test(
  log2(`Final probability correct`) ~ Final_Setting,
  data = filtered_data,
  paired = FALSE,
  conf.int = TRUE
)
```

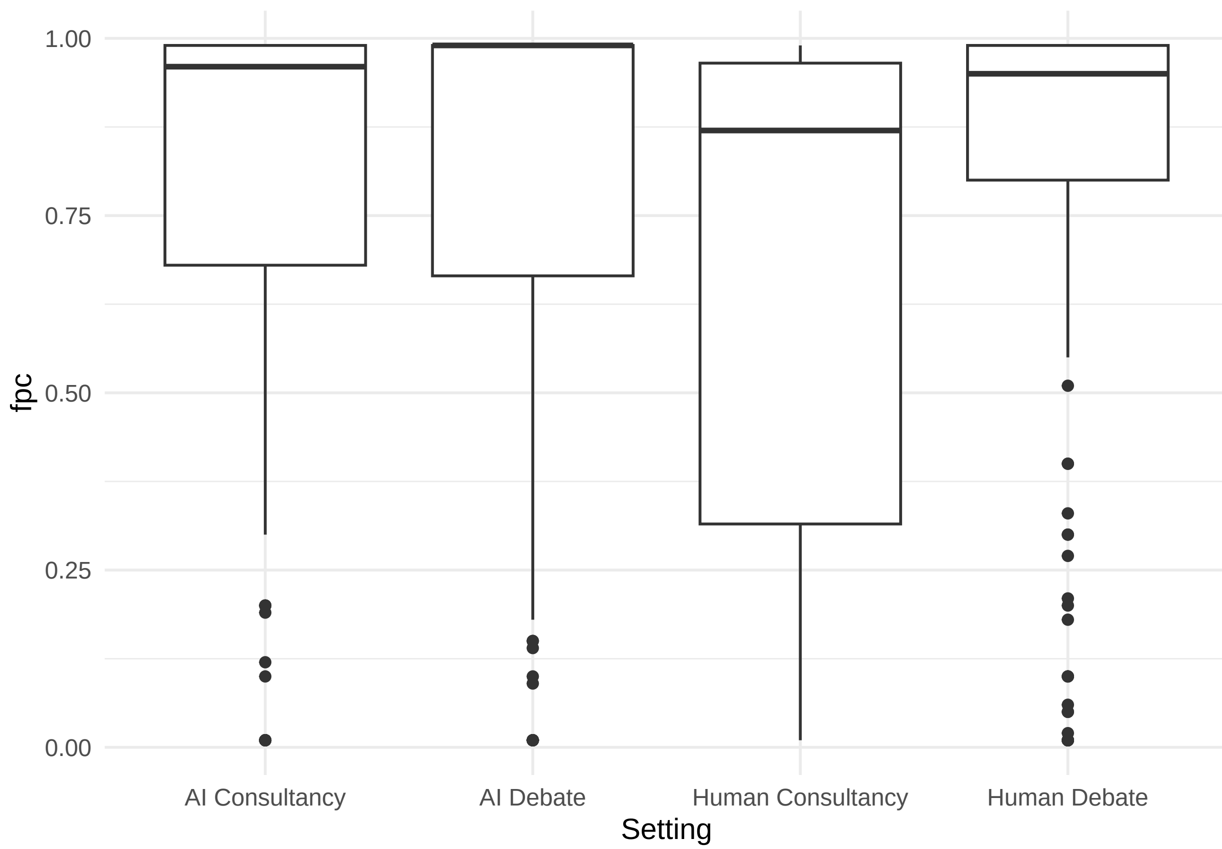
```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  log2(`Final probability correct`) by Final_Setting
## W = 6312.5, p-value = 0.001075
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -0.13752427127 -0.00002558317
## sample estimates:
## difference in location
##            -0.07801892
```

```r
# Conduct the Mann-Whitney U test and get the CI
wilcox_test(
  formula = `Final probability correct` ~ as.factor(Final_Setting),
  data = filtered_data,
  #weights = ~sampled_consultancies_all_debates_weights_grouped_setting,
  conf.int = TRUE  # Request the confidence interval
)
```

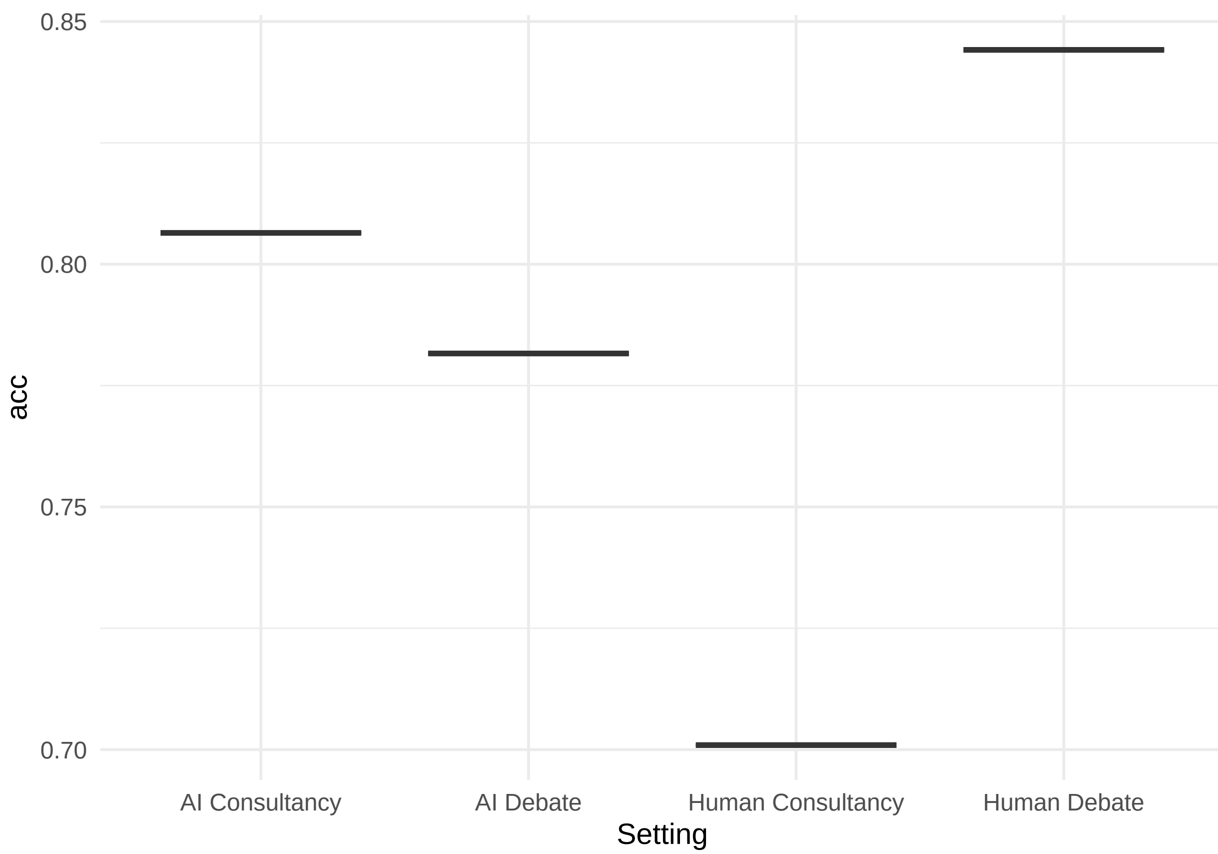
```
##
##  Asymptotic Wilcoxon-Mann-Whitney Test
##
## data:  Final probability correct by
```

```
##   as.factor(Final_Setting) (Human Consultancy, Human Debate)
## Z = -3.2776, p-value = 0.001047
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -0.090000004410453 -0.000000000314019
## sample estimates:
## difference in location
##                  -0.05
```

```r
# The rest is stuff i tried
judgments_online %>%
  ggplot() +
  geom_boxplot(aes(x = Final_Setting, y = fpc)) +
  labs(y = "fpc", x = "Setting")+
  theme_minimal()
```




```r
judgments_online %>%
  group_by(Final_Setting) %>% summarise(fpcmed = median(fpc),
                                        fpcmean = mean(Final_Accuracy)) %>%
  ggplot() +
  geom_boxplot(aes(x = Final_Setting, y = fpcmean)) +
  labs(y = "acc", x = "Setting")+
  theme_minimal()
```




```
consultancy_design <- svydesign(ids = ~1, data = subset(judgments_online, `Consultancy Sample` == TRUE


human_consultancy_design <- svydesign(ids = ~1, data = subset(judgments_online, `Human Consultancy Sampl


svyranktest(fpc~Final_Setting, human_consultancy_design)
```

```
##
##  Design-based KruskalWallis test
##
## data:  fpc ~ Final_Setting
## t = 1.5183, df = 248, p-value = 0.1302
## alternative hypothesis: true difference in mean rank score is not equal to 0
## sample estimates:
## difference in mean rank score
##                     0.0594665
```

```
judgments_online %>% group_by(Final_Setting) %>% summarise(fpcmed = median(fpc),
                                                          fpcmean = mean(fpc))
```

```
## # A tibble: 4 x 3
##   Final_Setting     fpcmed fpcmean
```

```
##   <chr>              <dbl>   <dbl>
## 1 AI Consultancy      0.96   0.764
## 2 AI Debate           0.99   0.754
## 3 Human Consultancy   0.87   0.672
## 4 Human Debate        0.95   0.794
```

```r
svyranktest(fpc~Final_Setting, consultancy_design, test = "median")
```

```
##
##  Design-based median test
##
## data:  fpc ~ Final_Setting
## df = 3, Chisq = 10.88, p-value = 0.01315
```

```r
svyranktest(fpc~Final_Setting, consultancy_design, test = "wilcoxon")
```

```
##
##  Design-based KruskalWallis test
##
## data:  fpc ~ Final_Setting
## df = 3, Chisq = 7.3067, p-value = 0.06431
```

```r
svyranktest(fpc~Final_Setting, consultancy_design, test = "vanderWaerden")
```

```
##
##  Design-based vanderWaerden test
##
## data:  fpc ~ Final_Setting
## df = 3, Chisq = 5.3917, p-value = 0.1471
```

```r
weighted_mannwhitney(fpc ~ Final_Setting + sampled_consultancies_all_debates_weights_grouped_setting, ju
```

```
## Warning in summary.glm(glm.object): observations with zero weight not used for
## calculating dispersion
```

```
##
## # Weighted Kruskal-Wallis test
##
##   comparison of fpc by Final_Setting
##   Chisq=3.00  df=7  p-value=0.063
```

```r
weighted_mannwhitney(fpc ~ Final_Setting + sampled_consultancies_all_debates_weights_grouped_setting, ju
```

```
## Warning in summary.glm(glm.object): observations with zero weight not used for
## calculating dispersion
```

```
##
## # Weighted Kruskal-Wallis test
##
##   comparison of fpc by Final_Setting
##   Chisq=3.00  df=7  p-value=0.063
```

Models

Logistic regression

```r
#judgments_online$Final_Setting <- relevel(judgments_online$Final_Setting, ref = "Human Debate")
model1 <- glm(Final_Accuracy ~ relevel(factor(Final_Setting), 'Human Debate'), family = 'binomial', data

summary(model1)
```

```
##
## Call:
## glm(formula = Final_Accuracy ~ relevel(factor(Final_Setting),
##     "Human Debate"), family = "binomial", data = judgments_online)
##
## Coefficients:
##                                                             Estimate
## (Intercept)                                                   1.6895
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy     -0.2624
## relevel(factor(Final_Setting), "Human Debate")AI Debate          -0.4144
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy  -0.8377
##                                                             Std. Error
## (Intercept)                                                     0.2222
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy        0.3439
## relevel(factor(Final_Setting), "Human Debate")AI Debate             0.3416
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy     0.3065
##                                                             z value
## (Intercept)                                                   7.604
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy     -0.763
## relevel(factor(Final_Setting), "Human Debate")AI Debate          -1.213
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy  -2.733
##                                                                      Pr(>|z|)
## (Intercept)                                                 0.0000000000000286
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy            0.44548
## relevel(factor(Final_Setting), "Human Debate")AI Debate                 0.22508
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy         0.00627
##
## (Intercept)                                                 ***
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy
## relevel(factor(Final_Setting), "Human Debate")AI Debate
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 454.34  on 440  degrees of freedom
## Residual deviance: 446.54  on 437  degrees of freedom
## AIC: 454.54
##
## Number of Fisher Scoring iterations: 4
```

```r
table(model1$fitted.values > 0.5)
```

```
##
## TRUE
##  441
```

```r
table(judgments_online$Final_Accuracy)
```

```
##
## FALSE  TRUE
##    93   348
```

```r
model2 <- glm(Final_Accuracy ~ relevel(factor(Participant),'Sean Wang') + relevel(factor(Final_Setting)

summary(model2)
```

```
##
## Call:
## glm(formula = Final_Accuracy ~ relevel(factor(Participant), "Sean Wang") +
##     relevel(factor(Final_Setting), "Human Debate"), family = "binomial",
##     data = judgments_online)
##
## Coefficients:
##                                                                 Estimate
## (Intercept)                                                       2.4207
## relevel(factor(Participant), "Sean Wang")Adelle Fernando         -0.9673
## relevel(factor(Participant), "Sean Wang")Aliyaah Toussaint       -0.1629
## relevel(factor(Participant), "Sean Wang")Anuj Jain               -1.0665
## relevel(factor(Participant), "Sean Wang")David Rein              -0.6335
## relevel(factor(Participant), "Sean Wang")Emmanuel Makinde       -17.9868
## relevel(factor(Participant), "Sean Wang")Ethan Rosen             -0.4748
## relevel(factor(Participant), "Sean Wang")Jackson Petty           -0.7391
## relevel(factor(Participant), "Sean Wang")Jessica Li              -0.3431
## relevel(factor(Participant), "Sean Wang")Julian Michael          -0.2693
## relevel(factor(Participant), "Sean Wang")Julien Dirani           13.1454
## relevel(factor(Participant), "Sean Wang")Max Layden              13.1454
## relevel(factor(Participant), "Sean Wang")Noor Mirza-Rashid       -1.5044
## relevel(factor(Participant), "Sean Wang")Reeya Kansra            -1.1296
## relevel(factor(Participant), "Sean Wang")Salsabila Mahdi         -0.4058
## relevel(factor(Participant), "Sean Wang")Sam Jin                 -0.1687
## relevel(factor(Participant), "Sean Wang")Shlomo Kofman           -1.2243
## relevel(factor(Participant), "Sean Wang")Shreeram Modi           -1.3599
## relevel(factor(Participant), "Sean Wang")Vishakh Padmakumar      -0.6289
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy     -0.3413
## relevel(factor(Final_Setting), "Human Debate")AI Debate          -0.4900
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy  -0.8203
##                                                                Std. Error
## (Intercept)                                                        0.5947
## relevel(factor(Participant), "Sean Wang")Adelle Fernando           0.7200
## relevel(factor(Participant), "Sean Wang")Aliyaah Toussaint         0.6793
## relevel(factor(Participant), "Sean Wang")Anuj Jain                 0.6336
## relevel(factor(Participant), "Sean Wang")David Rein                0.8453
```

```
## relevel(factor(Participant), "Sean Wang")Emmanuel Makinde       1455.3977
## relevel(factor(Participant), "Sean Wang")Ethan Rosen               1.2233
## relevel(factor(Participant), "Sean Wang")Jackson Petty             0.7360
## relevel(factor(Participant), "Sean Wang")Jessica Li                0.7235
## relevel(factor(Participant), "Sean Wang")Julian Michael            0.8322
## relevel(factor(Participant), "Sean Wang")Julien Dirani          1029.1216
## relevel(factor(Participant), "Sean Wang")Max Layden             1029.1216
## relevel(factor(Participant), "Sean Wang")Noor Mirza-Rashid         1.0265
## relevel(factor(Participant), "Sean Wang")Reeya Kansra              0.6831
## relevel(factor(Participant), "Sean Wang")Salsabila Mahdi           0.9594
## relevel(factor(Participant), "Sean Wang")Sam Jin                   0.6670
## relevel(factor(Participant), "Sean Wang")Shlomo Kofman             0.6211
## relevel(factor(Participant), "Sean Wang")Shreeram Modi             0.7215
## relevel(factor(Participant), "Sean Wang")Vishakh Padmakumar        1.2330
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy       0.3973
## relevel(factor(Final_Setting), "Human Debate")AI Debate            0.3972
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy    0.3702
##                                                                  z value
## (Intercept)                                                        4.070
## relevel(factor(Participant), "Sean Wang")Adelle Fernando          -1.344
## relevel(factor(Participant), "Sean Wang")Aliyaah Toussaint        -0.240
## relevel(factor(Participant), "Sean Wang")Anuj Jain                -1.683
## relevel(factor(Participant), "Sean Wang")David Rein               -0.749
## relevel(factor(Participant), "Sean Wang")Emmanuel Makinde         -0.012
## relevel(factor(Participant), "Sean Wang")Ethan Rosen              -0.388
## relevel(factor(Participant), "Sean Wang")Jackson Petty            -1.004
## relevel(factor(Participant), "Sean Wang")Jessica Li               -0.474
## relevel(factor(Participant), "Sean Wang")Julian Michael           -0.324
## relevel(factor(Participant), "Sean Wang")Julien Dirani             0.013
## relevel(factor(Participant), "Sean Wang")Max Layden                0.013
## relevel(factor(Participant), "Sean Wang")Noor Mirza-Rashid        -1.466
## relevel(factor(Participant), "Sean Wang")Reeya Kansra             -1.654
## relevel(factor(Participant), "Sean Wang")Salsabila Mahdi          -0.423
## relevel(factor(Participant), "Sean Wang")Sam Jin                  -0.253
## relevel(factor(Participant), "Sean Wang")Shlomo Kofman            -1.971
## relevel(factor(Participant), "Sean Wang")Shreeram Modi            -1.885
## relevel(factor(Participant), "Sean Wang")Vishakh Padmakumar       -0.510
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy      -0.859
## relevel(factor(Final_Setting), "Human Debate")AI Debate           -1.234
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy   -2.216
##                                                                  Pr(>|z|)
## (Intercept)                                                      0.000047 ***
## relevel(factor(Participant), "Sean Wang")Adelle Fernando           0.1791
## relevel(factor(Participant), "Sean Wang")Aliyaah Toussaint         0.8105
## relevel(factor(Participant), "Sean Wang")Anuj Jain                 0.0923 .
## relevel(factor(Participant), "Sean Wang")David Rein                0.4536
## relevel(factor(Participant), "Sean Wang")Emmanuel Makinde          0.9901
## relevel(factor(Participant), "Sean Wang")Ethan Rosen               0.6979
## relevel(factor(Participant), "Sean Wang")Jackson Petty             0.3153
## relevel(factor(Participant), "Sean Wang")Jessica Li                0.6353
## relevel(factor(Participant), "Sean Wang")Julian Michael            0.7462
## relevel(factor(Participant), "Sean Wang")Julien Dirani             0.9898
## relevel(factor(Participant), "Sean Wang")Max Layden                0.9898
## relevel(factor(Participant), "Sean Wang")Noor Mirza-Rashid         0.1428
```

```
## relevel(factor(Participant), "Sean Wang")Reeya Kansra            0.0982 .
## relevel(factor(Participant), "Sean Wang")Salsabila Mahdi         0.6723
## relevel(factor(Participant), "Sean Wang")Sam Jin                 0.8003
## relevel(factor(Participant), "Sean Wang")Shlomo Kofman           0.0487 *
## relevel(factor(Participant), "Sean Wang")Shreeram Modi           0.0595 .
## relevel(factor(Participant), "Sean Wang")Vishakh Padmakumar      0.6100
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy      0.3904
## relevel(factor(Final_Setting), "Human Debate")AI Debate           0.2173
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy   0.0267 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 454.34  on 440  degrees of freedom
## Residual deviance: 426.23  on 419  degrees of freedom
## AIC: 470.23
##
## Number of Fisher Scoring iterations: 14
```

```r
model3 <- glm(Final_Accuracy ~ relevel(factor(Participant),'Sean Wang') + relevel(factor(Final_Setting)

summary(model3)
```

```
##
## Call:
## glm(formula = Final_Accuracy ~ relevel(factor(Participant), "Sean Wang") +
##     relevel(factor(Final_Setting), "Human Debate") + `Untimed annotator context`,
##     family = "binomial", data = judgments_online)
##
## Coefficients:
##                                                                   Estimate
## (Intercept)                                                        2.39241
## relevel(factor(Participant), "Sean Wang")Adelle Fernando          -0.96852
## relevel(factor(Participant), "Sean Wang")Aliyaah Toussaint        -0.16398
## relevel(factor(Participant), "Sean Wang")Anuj Jain                -1.06685
## relevel(factor(Participant), "Sean Wang")David Rein               -0.63489
## relevel(factor(Participant), "Sean Wang")Emmanuel Makinde        -17.98846
## relevel(factor(Participant), "Sean Wang")Ethan Rosen              -0.47459
## relevel(factor(Participant), "Sean Wang")Jackson Petty            -0.74249
## relevel(factor(Participant), "Sean Wang")Jessica Li               -0.34530
## relevel(factor(Participant), "Sean Wang")Julian Michael           -0.27186
## relevel(factor(Participant), "Sean Wang")Julien Dirani            13.15230
## relevel(factor(Participant), "Sean Wang")Max Layden               13.14445
## relevel(factor(Participant), "Sean Wang")Noor Mirza-Rashid        -1.50376
## relevel(factor(Participant), "Sean Wang")Reeya Kansra             -1.13245
## relevel(factor(Participant), "Sean Wang")Salsabila Mahdi          -0.40553
## relevel(factor(Participant), "Sean Wang")Sam Jin                  -0.17101
## relevel(factor(Participant), "Sean Wang")Shlomo Kofman            -1.22652
## relevel(factor(Participant), "Sean Wang")Shreeram Modi            -1.36316
## relevel(factor(Participant), "Sean Wang")Vishakh Padmakumar       -0.62858
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy      -0.34156
## relevel(factor(Final_Setting), "Human Debate")AI Debate           -0.48986
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy   -0.82156
```

```
## `Untimed annotator context`                                          0.01124
##                                                                    Std. Error
## (Intercept)                                                           0.72828
## relevel(factor(Participant), "Sean Wang")Adelle Fernando              0.72028
## relevel(factor(Participant), "Sean Wang")Aliyaah Toussaint            0.67947
## relevel(factor(Participant), "Sean Wang")Anuj Jain                    0.63362
## relevel(factor(Participant), "Sean Wang")David Rein                   0.84559
## relevel(factor(Participant), "Sean Wang")Emmanuel Makinde          1455.39765
## relevel(factor(Participant), "Sean Wang")Ethan Rosen                  1.22336
## relevel(factor(Participant), "Sean Wang")Jackson Petty                0.73776
## relevel(factor(Participant), "Sean Wang")Jessica Li                   0.72417
## relevel(factor(Participant), "Sean Wang")Julian Michael               0.83307
## relevel(factor(Participant), "Sean Wang")Julien Dirani             1029.12003
## relevel(factor(Participant), "Sean Wang")Max Layden                1029.11021
## relevel(factor(Participant), "Sean Wang")Noor Mirza-Rashid            1.02656
## relevel(factor(Participant), "Sean Wang")Reeya Kansra                 0.68449
## relevel(factor(Participant), "Sean Wang")Salsabila Mahdi              0.95939
## relevel(factor(Participant), "Sean Wang")Sam Jin                      0.66793
## relevel(factor(Participant), "Sean Wang")Shlomo Kofman                0.62203
## relevel(factor(Participant), "Sean Wang")Shreeram Modi                0.72316
## relevel(factor(Participant), "Sean Wang")Vishakh Padmakumar           1.23306
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy          0.39741
## relevel(factor(Final_Setting), "Human Debate")AI Debate               0.39725
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy       0.37073
## `Untimed annotator context`                                           0.16727
##                                                                    z value
## (Intercept)                                                          3.285
## relevel(factor(Participant), "Sean Wang")Adelle Fernando            -1.345
## relevel(factor(Participant), "Sean Wang")Aliyaah Toussaint          -0.241
## relevel(factor(Participant), "Sean Wang")Anuj Jain                  -1.684
## relevel(factor(Participant), "Sean Wang")David Rein                 -0.751
## relevel(factor(Participant), "Sean Wang")Emmanuel Makinde           -0.012
## relevel(factor(Participant), "Sean Wang")Ethan Rosen                -0.388
## relevel(factor(Participant), "Sean Wang")Jackson Petty              -1.006
## relevel(factor(Participant), "Sean Wang")Jessica Li                 -0.477
## relevel(factor(Participant), "Sean Wang")Julian Michael             -0.326
## relevel(factor(Participant), "Sean Wang")Julien Dirani               0.013
## relevel(factor(Participant), "Sean Wang")Max Layden                  0.013
## relevel(factor(Participant), "Sean Wang")Noor Mirza-Rashid          -1.465
## relevel(factor(Participant), "Sean Wang")Reeya Kansra               -1.654
## relevel(factor(Participant), "Sean Wang")Salsabila Mahdi            -0.423
## relevel(factor(Participant), "Sean Wang")Sam Jin                    -0.256
## relevel(factor(Participant), "Sean Wang")Shlomo Kofman              -1.972
## relevel(factor(Participant), "Sean Wang")Shreeram Modi              -1.885
## relevel(factor(Participant), "Sean Wang")Vishakh Padmakumar         -0.510
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy        -0.859
## relevel(factor(Final_Setting), "Human Debate")AI Debate             -1.233
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy     -2.216
## `Untimed annotator context`                                          0.067
##                                                                    Pr(>|z|)
## (Intercept)                                                         0.00102 **
## relevel(factor(Participant), "Sean Wang")Adelle Fernando            0.17874
## relevel(factor(Participant), "Sean Wang")Aliyaah Toussaint          0.80929
## relevel(factor(Participant), "Sean Wang")Anuj Jain                  0.09223 .
```

```
## relevel(factor(Participant), "Sean Wang")David Rein             0.45276
## relevel(factor(Participant), "Sean Wang")Emmanuel Makinde       0.99014
## relevel(factor(Participant), "Sean Wang")Ethan Rosen            0.69806
## relevel(factor(Participant), "Sean Wang")Jackson Petty          0.31422
## relevel(factor(Participant), "Sean Wang")Jessica Li             0.63349
## relevel(factor(Participant), "Sean Wang")Julian Michael         0.74417
## relevel(factor(Participant), "Sean Wang")Julien Dirani          0.98980
## relevel(factor(Participant), "Sean Wang")Max Layden             0.98981
## relevel(factor(Participant), "Sean Wang")Noor Mirza-Rashid      0.14296
## relevel(factor(Participant), "Sean Wang")Reeya Kansra           0.09804 .
## relevel(factor(Participant), "Sean Wang")Salsabila Mahdi        0.67252
## relevel(factor(Participant), "Sean Wang")Sam Jin                0.79793
## relevel(factor(Participant), "Sean Wang")Shlomo Kofman          0.04863 *
## relevel(factor(Participant), "Sean Wang")Shreeram Modi          0.05943 .
## relevel(factor(Participant), "Sean Wang")Vishakh Padmakumar     0.61021
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy    0.39008
## relevel(factor(Final_Setting), "Human Debate")AI Debate         0.21753
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy 0.02669 *
## `Untimed annotator context`                                    0.94641
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 454.34  on 440  degrees of freedom
## Residual deviance: 426.23  on 418  degrees of freedom
## AIC: 472.23
##
## Number of Fisher Scoring iterations: 14
```

LMER

```
random.intercept.model = lmer(`Final probability correct` ~ (1|Final_Setting),
                              data = judgments, REML = TRUE)
judgments$random.intercept.preds = predict(random.intercept.model)
summary(random.intercept.model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: `Final probability correct` ~ (1 | Final_Setting)
##    Data: judgments
##
## REML criterion at convergence: 365.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5710 -0.2025  0.4765  0.5657  0.9501
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Final_Setting (Intercept) 0.003021 0.05497
##  Residual                  0.097613 0.31243
```

```
## Number of obs: 694, groups:  Final_Setting, 4
##
## Fixed effects:
##             Estimate Std. Error      df t value  Pr(>|t|)
## (Intercept)  0.75527    0.03071 3.29907   24.59 0.0000748 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
ranef(random.intercept.model)
```

```
## $Final_Setting
##                      (Intercept)
## AI Consultancy     0.0038517013
## AI Debate          0.0002838378
## Human Consultancy -0.0621214923
## Human Debate       0.0579859532
##
## with conditional variances for "Final_Setting"
```

```
ranova(random.intercept.model)
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## `Final probability correct` ~ (1 | Final_Setting)
##                      npar  logLik    AIC    LRT Df Pr(>Chisq)
## <none>                  3 -182.85 371.70
## (1 | Final_Setting)     2 -188.85 381.71 12.004  1  0.0005308 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
random.intercept.model = lmer(`Final probability correct` ~ (1|Participant) + (1|Final_Setting),
                              data = judgments, REML = TRUE)
judgments$random.intercept.preds = predict(random.intercept.model)
summary(random.intercept.model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: `Final probability correct` ~ (1 | Participant) + (1 | Final_Setting)
##    Data: judgments
##
## REML criterion at convergence: 359.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.7578 -0.1442  0.4283  0.6053  1.1245
##
## Random effects:
##  Groups        Name        Variance Std.Dev.
##  Participant   (Intercept) 0.002212 0.04703
##  Final_Setting (Intercept) 0.003078 0.05548
##  Residual                  0.095355 0.30880
```

```
## Number of obs: 694, groups:  Participant, 20; Final_Setting, 4
##
## Fixed effects:
##             Estimate Std. Error      df t value  Pr(>|t|)
## (Intercept)  0.75166    0.03341 4.27913    22.5 0.0000132 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
ranef(random.intercept.model)
```

```
## $Participant
##                      (Intercept)
## Adelle Fernando    -0.0220905028
## Aliyaah Toussaint   0.0471643746
## Anuj Jain          -0.0445616201
## David Rein          0.0114523799
## Emmanuel Makinde   -0.0115800001
## Ethan Rosen        -0.0169492748
## Jackson Petty      -0.0043494488
## Jessica Li         -0.0037154915
## Julian Michael      0.0358936006
## Julien Dirani      -0.0007219769
## Max Layden         -0.0038070795
## Noor Mirza-Rashid  -0.0116092402
## Reeya Kansra       -0.0239989270
## Salsabila Mahdi     0.0325053849
## Sam Arnesen        -0.0219751214
## Sam Jin             0.0507287062
## Sean Wang           0.0488780322
## Shlomo Kofman      -0.0467944429
## Shreeram Modi       0.0032383189
## Vishakh Padmakumar -0.0177076712
##
## $Final_Setting
##                    (Intercept)
## AI Consultancy     0.0025422646
## AI Debate          0.0003481343
## Human Consultancy -0.0621094626
## Human Debate       0.0592190637
##
## with conditional variances for "Participant" "Final_Setting"
```

```r
ranova(random.intercept.model)
```

```
## ANOVA-like table for random-effects: Single term deletions
##
## Model:
## `Final probability correct` ~ (1 | Participant) + (1 | Final_Setting)
##                     npar  logLik    AIC     LRT Df Pr(>Chisq)
## <none>                 4 -179.86 367.72
## (1 | Participant)      3 -182.85 371.70  5.9769  1  0.0144944 *
## (1 | Final_Setting)    3 -185.43 376.86 11.1371  1  0.0008462 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

BRMS

```
#brm1 <- brm(data = judgments_online,
#             formula = as.numeric(Final_Accuracy) | trials(2) ~ 1 + (1 | Final_Setting),
#             family = binomial("identity"),
#             iter = 2000, warmup = 1000, chains = 4, cores = 4,
#             control = list(adapt_delta = .975, max_treedepth = 20),
#             seed = 190831)
#plot(brm1)
```

Efficiency

Quotes %, caveats

```
debater_turns = turns.merge(
        judgments_online[["Room name", "Question", "Story length",
                 "Untimed annotator context","Untimed annotator context bins",
                 "Setting", "Final_Setting", "Final_Accuracy",
                 "Is offline","Number of judge continues","Participant"]],
        how="inner",
        on="Room name",
    )



# Filtering for specific roles
debater_turns = debater_turns[debater_turns['Role (honest/dishonest)'].isin(['Honest debater', 'Dishone

# Aggregating function to concatenate quote spans
def custom_join(series):
    return ' '.join(filter(lambda x: isinstance(x, str), series))


aggregates = {
    'Quote length': 'sum',
    'Story length': 'mean',
    'Number of judge continues': 'max',
    'Participant quote span': custom_join
}
debater_turns_agg = debater_turns.groupby('Room name').agg(aggregates).reset_index()
debater_turns_agg_simple = debater_turns_agg.merge(
    debater_turns[['Room name', 'Setting', 'Final_Setting', 'Question', 'Untimed annotator context bins
    on='Room name'
)


# Extracting the spans
def extract_spans(span_str):
    """Extract numerical spans from the given string."""
    if pd.isna(span_str):
        return []
```

```
    spans = re.findall(r'<<(\d+)-(\d+)>>', span_str)
    return [(int(start), int(end)) for start, end in spans]

# Functions to compute and compare spans across settings
def extract_numbers_from_span(span_str):
    spans = extract_spans(span_str)
    numbers = set()
    for start, end in spans:
        numbers.update(range(int(start), int(end)+1))
    return numbers
def quote_length(span_str):
  spans = extract_spans(span_str)
  numbers = set()
  for start, end in spans:
    numbers.update(range(int(start), int(end)))
  return numbers

# Merging overlapping spans
def merge_overlapping_spans(span_str):
    if not isinstance(span_str, str):
        return span_str
    spans = extract_spans(span_str)
    if not spans:
        return span_str
    spans.sort(key=lambda x: x[0])
    merged = [spans[0]]
    for current in spans:
        previous = merged[-1]
        if current[0] <= previous[1]:
            upper_bound = max(previous[1], current[1])
            merged[-1] = (previous[0], upper_bound)
        else:
            merged.append(current)
    return ' '.join(f'<<{start}-{end}>>' for start, end in merged)


debater_turns_agg_simple["quote_length"] = debater_turns_agg_simple["Participant quote span"].apply(laml




# Identify questions with more than one setting and filter out the debater_turns dataframe
questions_with_multi_settings = debater_turns.groupby("Question").filter(lambda x: len(x["Setting"].unic
debater_turns_filtered = debater_turns[debater_turns["Question"].isin(questions_with_multi_settings)]

# Aggregating data
aggregates = {
    'Quote length': 'sum',
    'Story length': 'mean',
    'Number of judge continues': 'max',
    'Participant quote span': custom_join
}
# Grouping by 'Room name' and aggregating
```

```python
debater_turns_filtered_by_room = debater_turns_filtered.groupby('Room name').agg(aggregates).reset_inde

# Merging the aggregated results with the original data to reintroduce the desired columns
debater_turns_agg = debater_turns_filtered_by_room.merge(
    debater_turns_filtered[['Room name', 'Setting', 'Final_Setting', 'Question', 'Untimed annotator con
    on='Room name'
)

debater_turns_agg["quote_length"] = debater_turns_agg["Participant quote span"].apply(lambda row: len(q

# Merge overlapping spans after the aggregation
debater_turns_agg["merged_quote_spans"] = debater_turns_agg["Participant quote span"].apply(merge_overla

#debater_turns_agg["merged_quote_length"] = debater_turns_agg["Participant quote span"].apply(lambda ro
#print(debater_turns_agg["merged_quote_length"][1])
#print((debater_turns_agg["merged_quote_length"]==debater_turns_agg["quote_length"]).value_counts())

#print((debater_turns_agg['quote_length'].fillna(0)/debater_turns_agg['Story length'].fillna(0)).descri


def convert_to_span_format(numbers):
    sorted_numbers = sorted(list(numbers))
    spans = []
    if sorted_numbers:
        start = sorted_numbers[0]
        end = sorted_numbers[0]
        for num in sorted_numbers[1:]:
            if num == end + 1:
                end = num
            else:
                spans.append((start, end))
                start = end = num
        spans.append((start, end))
    return ' '.join(f'<<{start}-{end}>>' for start, end in spans)

def compute_span_differences(dataframe):
    differences = {}
    for question, group in dataframe.groupby("Question"):
        settings = group["Setting"].unique()
        if len(settings) > 1:
            for i in range(len(settings)):
                for j in range(i+1, len(settings)):
                    setting_1 = settings[i]
                    setting_2 = settings[j]
                    room_1 = group[group["Setting"] == setting_1]["Room name"].values[0]
                    room_2 = group[group["Setting"] == setting_2]["Room name"].values[0]
                    acc_1 = group[group["Setting"] == setting_1]["Final_Accuracy"].values[0]
                    acc_2 = group[group["Setting"] == setting_2]["Final_Accuracy"].values[0]
                    span_str_1 = group[group["Setting"] == setting_1]["merged_quote_spans"].values[0]
                    span_str_2 = group[group["Setting"] == setting_2]["merged_quote_spans"].values[0]
                    numbers_1 = extract_numbers_from_span(span_str_1)
                    numbers_2 = extract_numbers_from_span(span_str_2)
                    diff_1 = numbers_1 - numbers_2
```

```
                    diff_2 = numbers_2 - numbers_1
                    key = (question, setting_1, room_1, acc_1, setting_2, room_2, acc_2)
                    value = (convert_to_span_format(diff_1), convert_to_span_format(diff_2))
                    differences[key] = value
    return differences

span_differences_all = compute_span_differences(debater_turns_agg)

#print(span_differences_all.keys())
#for span in span_differences_all[('Why were Jorgenson and Ganti not put to death?', 'Human Consultancy
#  print(len(quote_length(span)))
```

```
split_span_differences_with_room = []
# Iterate over the span differences
for (question, setting_1, room_1, acc_1, setting_2, room_2, acc_2), (diff_1, diff_2) in span_differences
    split_span_differences_with_room.append((question, setting_1, room_1, acc_1, setting_2, room_2, acc_
    split_span_differences_with_room.append((question, setting_2, room_2, acc_2, setting_1, room_1, acc_

# Convert the list to a DataFrame
split_span_df = pd.DataFrame(split_span_differences_with_room, columns=['Question', 'Setting 1', 'Room 1

split_span_df["Span Difference Count"] = split_span_df["Span Difference"].apply(lambda x: len(quote_leng
split_span_df["Settings"] = split_span_df["Setting 1"] + " - " + split_span_df["Setting 2"]


# Group by the new 'Settings' column and compute aggregated counts and average of 'Span Difference Coun
grouped_data = split_span_df.groupby("Settings").agg(
    Count=('Span Difference Count', 'size'),
    Average_Span_Difference=('Span Difference Count', 'mean')
).reset_index()

grouped_data
```

```
##                                               Settings  ...  Average_Span_Difference
## 0    AI Consultancy Dishonest - AI Consultancy Honest  ...               138.909091
## 1                 AI Consultancy Dishonest - AI Debate  ...               142.818182
## 2   AI Consultancy Dishonest - Human Consultancy D...  ...               154.000000
## 3   AI Consultancy Dishonest - Human Consultancy H...  ...                83.285714
## 4              AI Consultancy Dishonest - Human Debate  ...               103.555556
## 5    AI Consultancy Honest - AI Consultancy Dishonest  ...               202.818182
## 6                    AI Consultancy Honest - AI Debate  ...               189.750000
## 7   AI Consultancy Honest - Human Consultancy Dish...  ...               206.900000
## 8    AI Consultancy Honest - Human Consultancy Honest  ...               163.666667
## 9                 AI Consultancy Honest - Human Debate  ...               198.222222
## 10                AI Debate - AI Consultancy Dishonest  ...                79.454545
## 11                   AI Debate - AI Consultancy Honest  ...                65.500000
## 12             AI Debate - Human Consultancy Dishonest  ...                95.500000
## 13                AI Debate - Human Consultancy Honest  ...                78.666667
## 14                         AI Debate - Human Debate  ...                88.272727
## 15  Human Consultancy Dishonest - AI Consultancy D...  ...               350.300000
## 16  Human Consultancy Dishonest - AI Consultancy H...  ...               313.200000
## 17             Human Consultancy Dishonest - AI Debate  ...               426.100000
## 18  Human Consultancy Dishonest - Human Consultanc...  ...               321.100000
```

```
## 19         Human Consultancy Dishonest - Human Debate  ...               311.684211
## 20  Human Consultancy Honest - AI Consultancy Dish...  ...               248.714286
## 21   Human Consultancy Honest - AI Consultancy Honest  ...               311.333333
## 22               Human Consultancy Honest - AI Debate  ...               298.333333
## 23  Human Consultancy Honest - Human Consultancy D...  ...               265.133333
## 24            Human Consultancy Honest - Human Debate  ...               259.750000
## 25            Human Debate - AI Consultancy Dishonest  ...               205.444444
## 26               Human Debate - AI Consultancy Honest  ...               221.444444
## 27                           Human Debate - AI Debate  ...               205.636364
## 28         Human Debate - Human Consultancy Dishonest  ...               160.552632
## 29            Human Debate - Human Consultancy Honest  ...               145.071429
##
## [30 rows x 3 columns]
```

```
filtered_df = split_span_df[
    (split_span_df["Setting 1"] == "Human Debate") &
    ((split_span_df["Setting 2"] == "Human Consultancy Honest") | (split_span_df["Setting 2"] == "Human
]

print(filtered_df.groupby(['Setting 2','Acc_1','Acc_2'])['Span Difference Count'].describe())
```

```
##                                          count        mean  ...    75%    max
## Setting 2                   Acc_1 Acc_2                     ...
## Human Consultancy Dishonest False False    3.0  129.333333  ...  138.0  145.0
##                                   True     3.0  135.333333  ...  183.5  275.0
##                             True  False   15.0  140.933333  ...  179.0  254.0
##                                   True    17.0  187.823529  ...  225.0  526.0
## Human Consultancy Honest    False True     6.0  117.333333  ...  139.5  269.0
##                             True  False    1.0  120.000000  ...  120.0  120.0
##                                   True    21.0  154.190476  ...  200.0  394.0
##
## [7 rows x 8 columns]
```

```
# Calculate the IQR and bounds for each group in 'Setting 2'
grouped = filtered_df.groupby('Setting 2')['Span Difference Count']

Q1 = grouped.quantile(0.25)
Q3 = grouped.quantile(0.75)
IQR = Q3 - Q1

lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Filter out the outliers based on the computed bounds
filtered_no_outliers = filtered_df[
    (filtered_df['Setting 2'].map(lower_bound) <= filtered_df['Span Difference Count']) &
    (filtered_df['Setting 2'].map(upper_bound) >= filtered_df['Span Difference Count'])
]

filtered_no_outliers
```

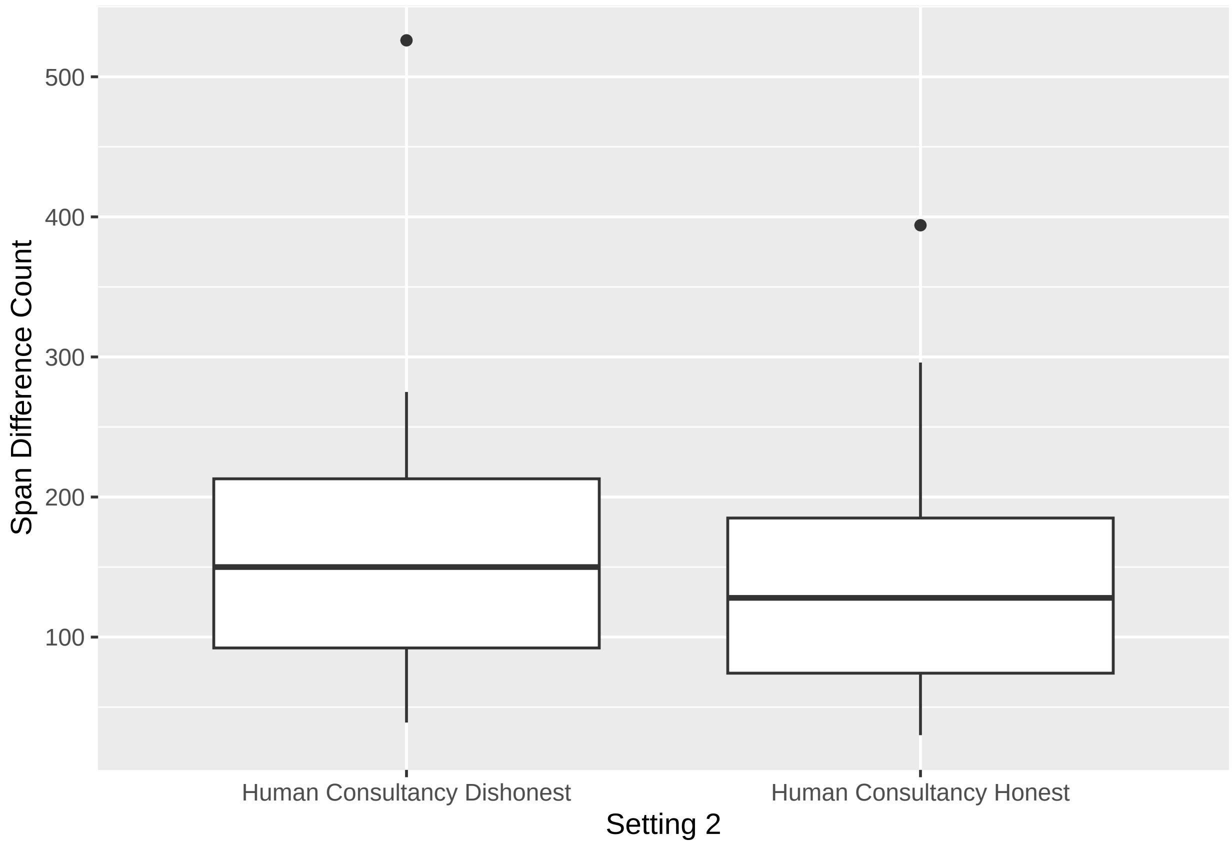
```
##                                               Question  ...                                   Settin
## 0    By the end of the passage. what can we underst...  ...  Human Debate - Human Consultancy Dishon
```

```
## 20   Did the questions Tremaine needed answers to g...  ...  Human Debate - Human Consultancy Dishone
## 40   From the information the story provides, do yo...  ...     Human Debate - Human Consultancy Hone
## 64          How did Hendricks outfit the ship for war?  ...     Human Debate - Human Consultancy Hone
## 66          How did Hendricks outfit the ship for war?  ...  Human Debate - Human Consultancy Dishone
## ..                                                ...  ...
## 384  Why was the main character daydreaming about b...  ...     Human Debate - Human Consultancy Hone
## 386  Why was the main character daydreaming about b...  ...  Human Debate - Human Consultancy Dishone
## 390            Why was the murderer trying to kill Bo?  ...  Human Debate - Human Consultancy Dishone
## 410     Why were Jorgenson and Ganti not put to death?  ...  Human Debate - Human Consultancy Dishone
## 412     Why were Jorgenson and Ganti not put to death?  ...     Human Debate - Human Consultancy Hone
##
## [64 rows x 10 columns]
```

```
print(filtered_no_outliers.groupby(['Setting 2','Acc_1','Acc_2'])['Span Difference Count'].describe())
```

```
##                                         count        mean  ...     75%    max
## Setting 2                   Acc_1 Acc_2                    ...
## Human Consultancy Dishonest False False    3.0  129.333333  ...  138.00  145.0
##                                   True     3.0  135.333333  ...  183.50  275.0
##                             True  False   15.0  140.933333  ...  179.00  254.0
##                                   True    16.0  166.687500  ...  221.25  266.0
## Human Consultancy Honest    False True     6.0  117.333333  ...  139.50  269.0
##                             True  False    1.0  120.000000  ...  120.00  120.0
##                                   True    20.0  142.200000  ...  185.00  296.0
##
## [7 rows x 8 columns]
```

```
debater_turns<- py$debater_turns_agg_simple
debater_turns$check <- paste0(debater_turns$Participant_y, debater_turns$`Room name`)
sample.rooms <- read.csv("~/Downloads/sample-rooms-2.csv", header=FALSE)
sample.rooms_samples <- sort(paste0(sample.rooms$V2, sample.rooms$V1))
debater_turns <- subset(debater_turns, debater_turns$check %in% sample.rooms_samples)

span_difference_debate_consultancies<-py$filtered_df
ggplot(span_difference_debate_consultancies) +
  geom_boxplot(aes(x = `Setting 2`, y = `Span Difference Count`))
```

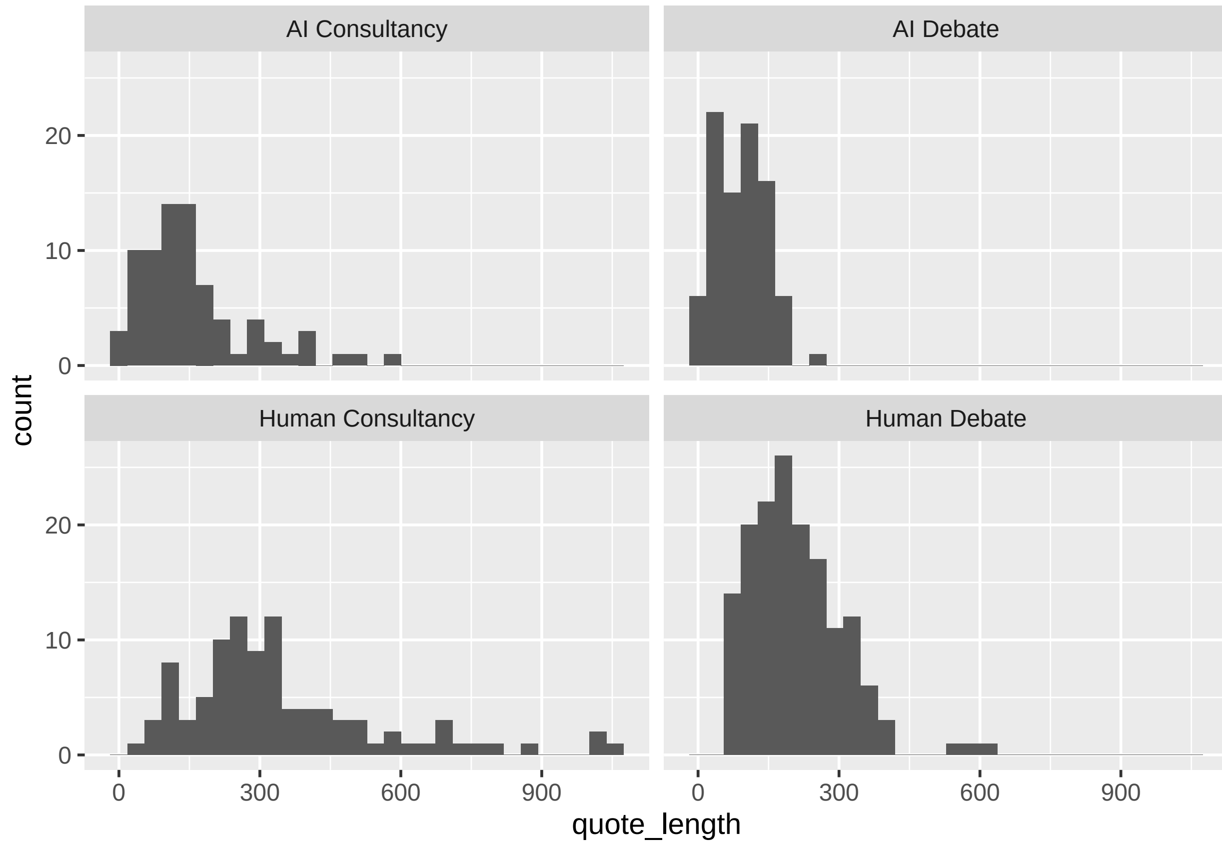



```
final_table_desc_stats <- debater_turns %>% group_by(Final_Setting) %>% summarise(n = n(), rounds = mean
knitr::kable(final_table_desc_stats, booktab = TRUE, digits = 1, col.names = c("Setting", "n", "rounds p
```

Setting	n	rounds per debate	quoted tokens per debate	tokens per round
AI Consultancy	76	4.2	158.6	45.2
AI Debate	87	3.8	90.5	28.4
Human Consultancy	96	4.0	347.2	87.8
Human Debate	154	2.7	208.2	76.5

```
filtered_outliers <- debater_turns %>%
  group_by(Final_Setting) %>%
  mutate(Q1 = quantile(quote_length, 0.25),
         Q3 = quantile(quote_length, 0.75),
         IQR = Q3 - Q1,
         lower_bound = Q1 - 1.5 * IQR,
         upper_bound = Q3 + 1.5 * IQR)

ggplot(data = debater_turns) +
  geom_histogram(aes(x = quote_length, binwidth = 1)) +
  facet_wrap(~ Final_Setting)
```

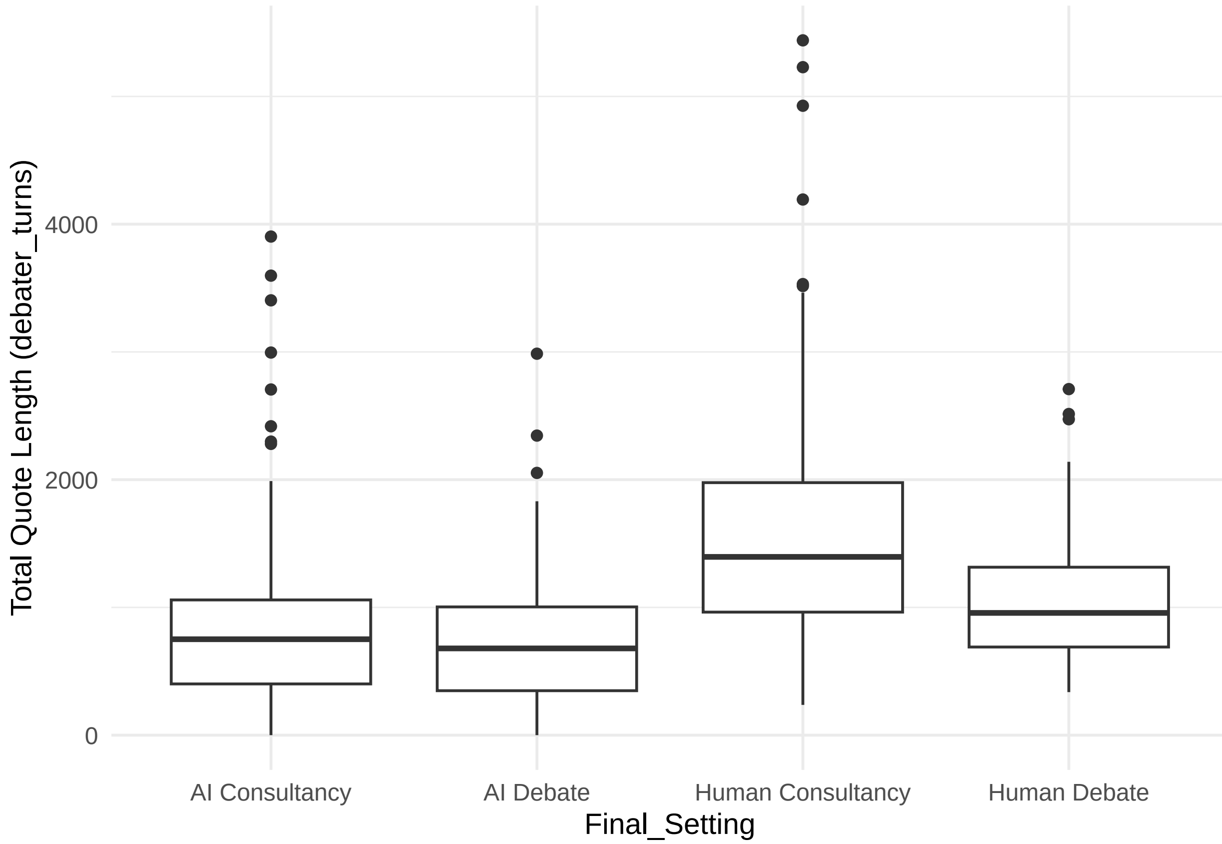
```
## Warning in geom_histogram(aes(x = quote_length, binwidth = 1)): Ignoring
## unknown aesthetics: binwidth

## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



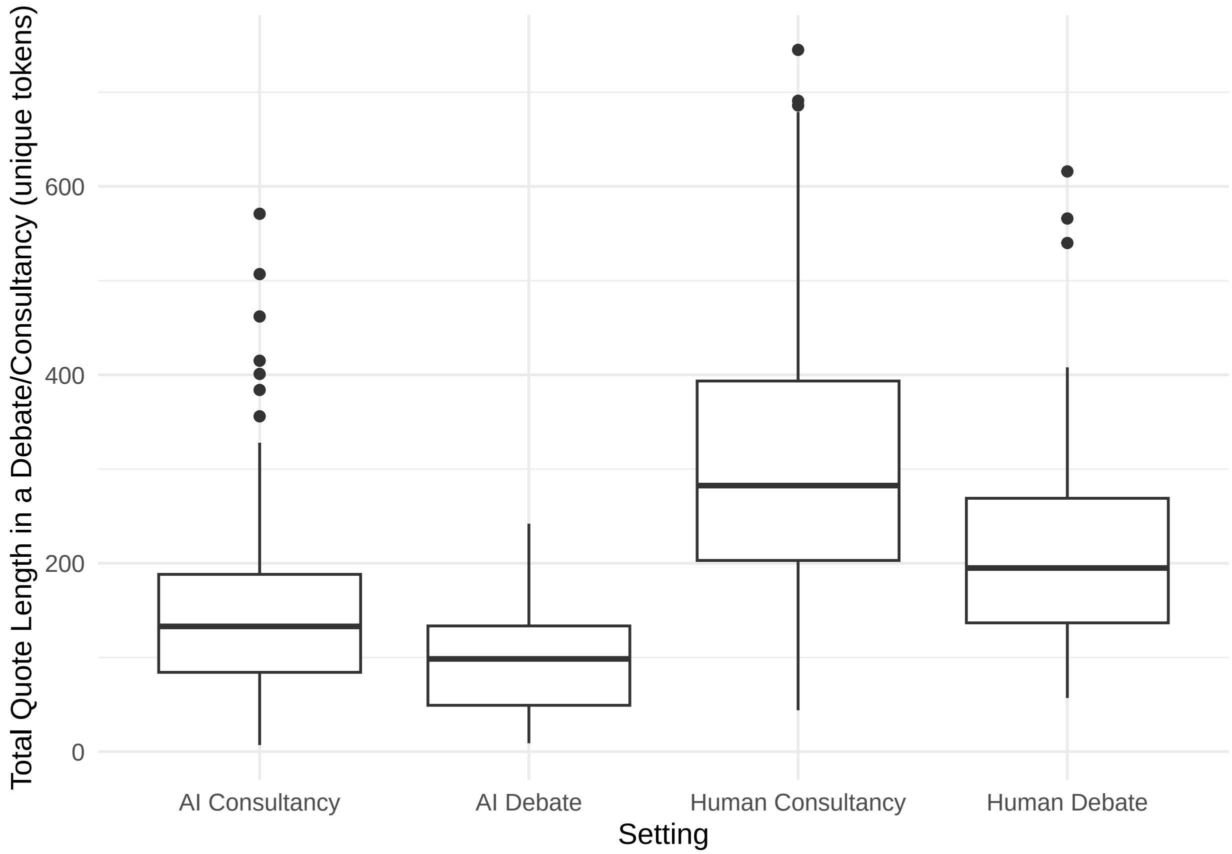

```
pairwise.t.test(debater_turns$quote_length, debater_turns$Final_Setting)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  debater_turns$quote_length and debater_turns$Final_Setting
##
##                   AI Consultancy       AI Debate            Human Consultancy
## AI Debate         0.0024               -                    -
## Human Consultancy < 0.0000000000000002 < 0.0000000000000002 -
## Human Debate      0.0080               0.000000000366633    0.000000000000036
##
## P value adjustment method: holm
```

```
ggplot(debater_turns) +
  geom_boxplot(aes(x = Final_Setting, y = `Quote length`)) +
  labs(y = "Total Quote Length (debater_turns)")+
  theme_minimal()
```

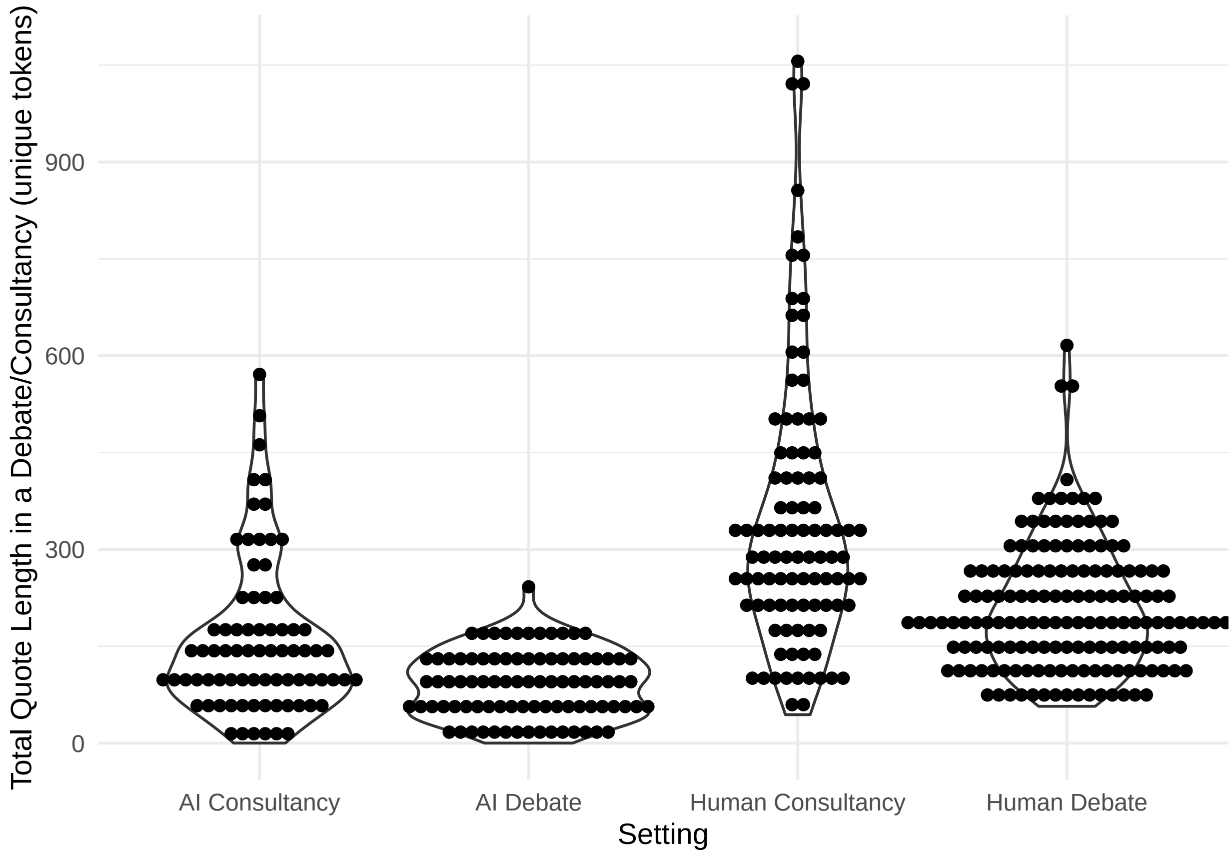



```r
filtered <- debater_turns %>%
  group_by(Final_Setting) %>%
  mutate(Q1 = quantile(quote_length, 0.25),
         Q3 = quantile(quote_length, 0.75),
         IQR = Q3 - Q1,
         lower_bound = Q1 - 1.5 * IQR,
         upper_bound = Q3 + 1.5 * IQR) %>%
  filter(quote_length > 0 & quote_length < 750) %>%
  select(-Q1, -Q3, -IQR, -lower_bound, -upper_bound)
filtered %>%
  ggplot() +
  geom_boxplot(aes(x = Final_Setting, y = quote_length)) +
  labs(y = "Total Quote Length in a Debate/Consultancy (unique tokens)", x = "Setting")+
  theme_minimal()
```



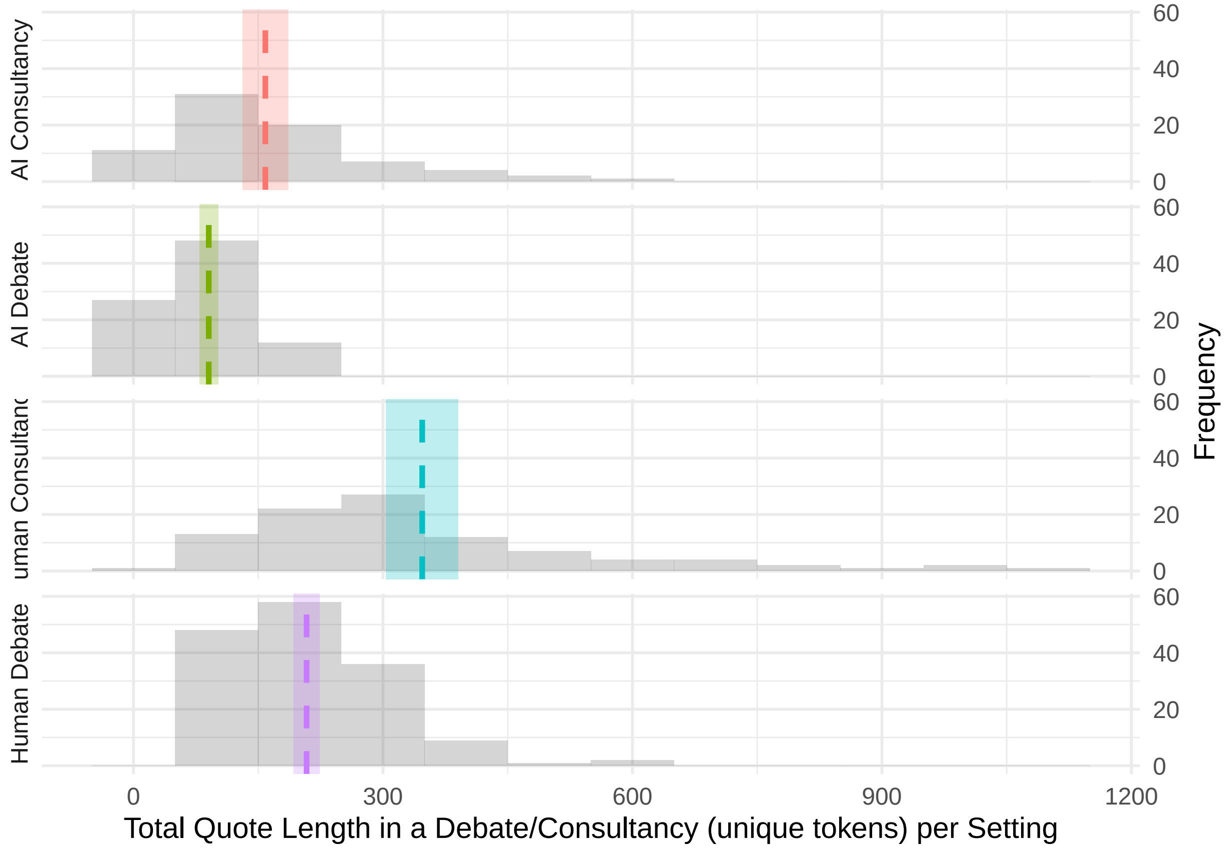

```
debater_turns %>%
  ggplot() +
  geom_violin(aes(x = Final_Setting, y = quote_length)) +
  geom_dotplot(aes(x = Final_Setting, y = quote_length), binaxis= "y",
               stackdir = "center",
               dotsize = 0.5,
               fill = 1) +
  labs(y = "Total Quote Length in a Debate/Consultancy (unique tokens)", x = "Setting")+
  theme_minimal()
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```




```
debater_turns %>%
  group_by(Final_Setting) %>%
  summarise(avg = mean(quote_length),
            lower_ci = t.test(quote_length)$conf.int[1],
            upper_ci = t.test(quote_length)$conf.int[2]) %>%
  ggplot(aes(x = avg)) +
  geom_histogram(data = debater_turns, aes(x = quote_length), binwidth = 100, alpha = 0.25) +
  geom_vline(aes(xintercept = avg, color = Final_Setting), linetype="dashed", size=1) +
  geom_rect(aes(xmin = lower_ci, xmax = upper_ci, ymin = -Inf, ymax = Inf, fill = Final_Setting), alpha
  labs(x = "Total Quote Length in a Debate/Consultancy (unique tokens) per Setting",
       y = "Frequency") +
  facet_wrap(~Final_Setting, ncol = 1, strip.position = "left") +
  theme_minimal() +
  theme(
    axis.title.y.right = element_text(angle = 90),
  ) +
  scale_y_continuous(position = "right") +
  theme(legend.position="none")
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```
pairwise.t.test(filtered$quote_length, filtered$Final_Setting)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  filtered$quote_length and filtered$Final_Setting
##
##                  AI Consultancy      AI Debate            Human Consultancy
## AI Debate        0.00044             -                    -
## Human Consultancy 0.0000000000000062 < 0.0000000000000002 -
## Human Debate     0.00438             0.0000000000049861   0.0000000001106057
##
## P value adjustment method: holm
```

```
filtered %>% group_by(Final_Setting) %>% summarise(avground = median(quote_length))
```

```
## # A tibble: 4 x 2
##   Final_Setting     avground
##   <chr>                <dbl>
## 1 AI Consultancy       133
## 2 AI Debate             98.5
## 3 Human Consultancy    282.
## 4 Human Debate         195
```

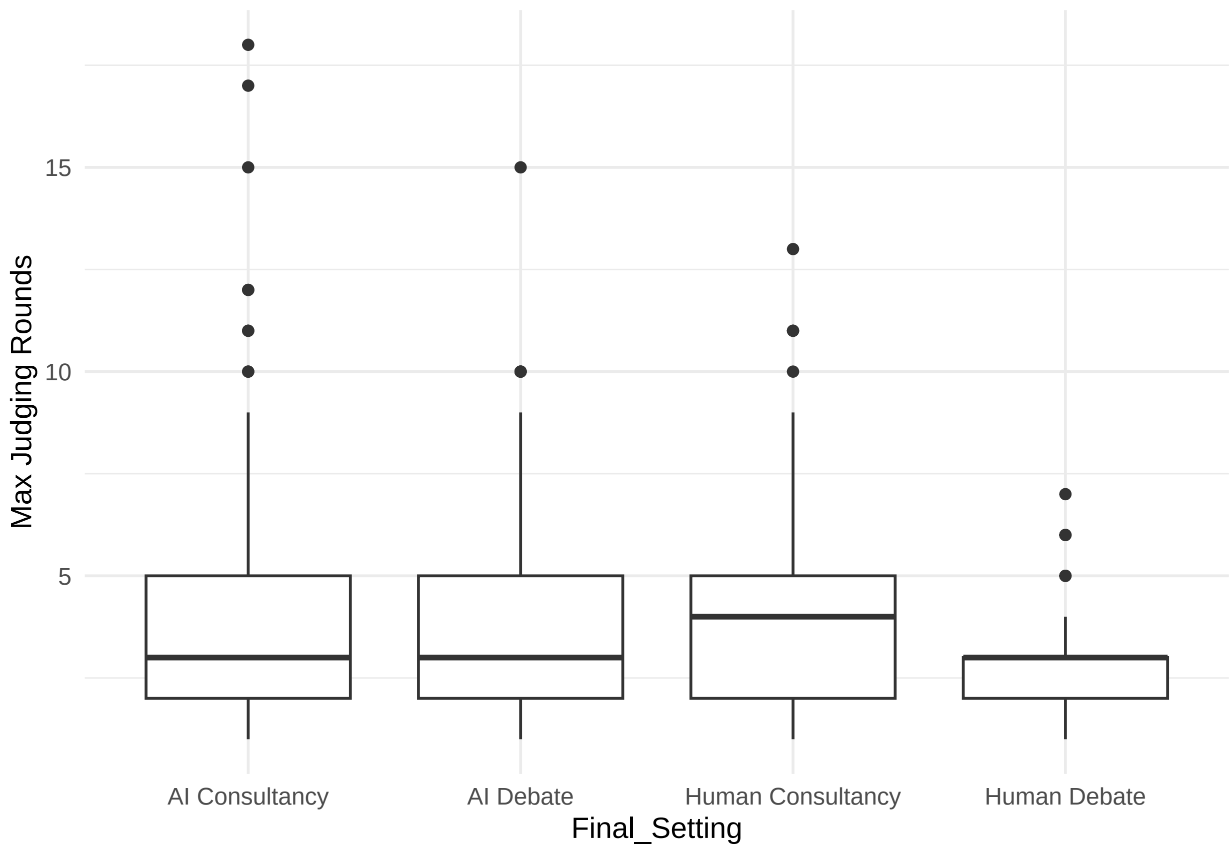
```r
debater_turns %>% group_by(Final_Setting) %>% summarise(avground = median(quote_length))
```

```
## # A tibble: 4 x 2
##   Final_Setting     avground
##   <chr>                <dbl>
## 1 AI Consultancy        130
## 2 AI Debate              92
## 3 Human Consultancy     294.
## 4 Human Debate          195
```

```r
debater_turns <- debater_turns %>%
  group_by(`Room name`) %>%
  mutate(`Max judge rounds by room` = max(`Number of judge continues`, na.rm = TRUE)) %>%
  ungroup()
debater_turns <- debater_turns %>%
  mutate(`Max judge rounds bin` = factor(ifelse(`Max judge rounds by room` > 7, "8", as.character(`Max j

table(debater_turns$`Max judge rounds bin`)
```

```
##
##   1   2   3   4   5   6   7   8
##  59  95 107  72  28  11   9  32
```

```r
ggplot(debater_turns) +
  geom_boxplot(aes(x = Final_Setting, y = `Max judge rounds by room`)) +
  labs(y = 'Max Judging Rounds') +
  theme_minimal()
```




```
pairwise.t.test(debater_turns$`Max judge rounds by room`, debater_turns$Final_Setting)
```

```
##
##  Pairwise comparisons using t tests with pooled SD
##
## data:  debater_turns$'Max judge rounds by room' and debater_turns$Final_Setting
##
##                   AI Consultancy AI Debate Human Consultancy
## AI Debate         1.00000        -         -
## Human Consultancy 1.00000        1.00000   -
## Human Debate      0.00020        0.00482   0.00037
##
## P value adjustment method: holm
```

```
table(round(debater_turns$quote_length, -2))
```

```
##
##    0  100  200  300  400  500  600  700  800  900 1000 1100
##   39  140  112   70   25   10    7    4    2    1    2    1
```

```
debater_turns$quote_length_bin <- as.factor(round(debater_turns$quote_length, -2))
debater_turns$quote_length_bin <- ordered(debater_turns$quote_length_bin, levels = paste(sort(as.intege
table(debater_turns$quote_length_bin)
```

```
##
##    0  100  200  300  400  500  600  700  800  900 1000 1100
##   39  140  112   70   25   10    7    4    2    1    2    1
```

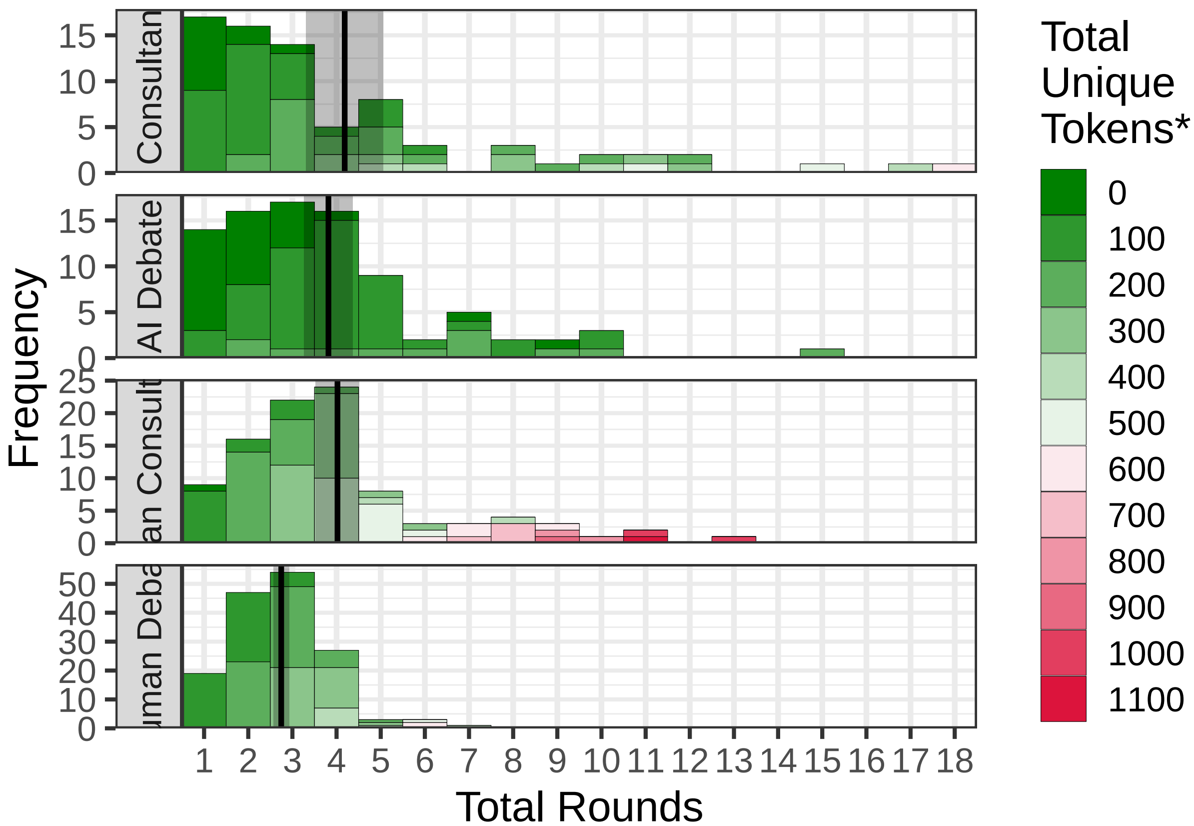
```r
# Define the color function and palette
colfunc <- colorRampPalette(c(correctColor,"white",incorrectColor))
palette <- colfunc(length(levels(as.factor(debater_turns$`Max judge rounds bin`))))

# Plot
debater_turns %>%
  filter(`Max judge rounds bin` != "0") %>%
  group_by(Final_Setting) %>%
  summarise(avg = mean(as.numeric(levels(quote_length_bin))[quote_length_bin], na.rm = T),
            lower_ci = t.test(as.numeric(levels(quote_length_bin))[quote_length_bin], na.rm = T)$conf.i
            upper_ci = t.test(as.numeric(levels(quote_length_bin))[quote_length_bin], na.rm = T)$conf.i
            n = n())
```

```
## # A tibble: 4 x 5
##   Final_Setting       avg lower_ci upper_ci     n
##   <chr>             <dbl>    <dbl>    <dbl> <int>
## 1 AI Consultancy    163.     134.     192.     76
## 2 AI Debate          82.8     68.9     96.6    87
## 3 Human Consultancy 343.     298.     387.     96
## 4 Human Debate      211.     195.     227.    154
```

```r
debater_turns %>%
  filter(`Max judge rounds bin` != "0" & !is.na(Final_Setting)) %>%
  group_by(Final_Setting) %>%
  summarise(avg = mean(as.numeric(levels(quote_length_bin))[quote_length_bin], na.rm = T),
            lower_ci = t.test(as.numeric(levels(quote_length_bin))[quote_length_bin], na.rm = T)$conf.i
            upper_ci = t.test(as.numeric(levels(quote_length_bin))[quote_length_bin], na.rm = T)$conf.i
  ggplot(aes(x = avg)) +
  geom_bar(data = debater_turns %>% filter(`Max judge rounds bin` != "0" & !is.na(Final_Setting)),
                aes(x = as.numeric(levels(quote_length_bin))[quote_length_bin], fill = as.factor(`Max j
                position='stack',
                color = "black",
                size = 0.1) +
  #geom_vline(aes(xintercept = avg), size=1, color = "black") +
  #geom_rect(aes(xmin = lower_ci, xmax = upper_ci, ymin = -Inf, ymax = Inf), alpha = 0.25, fill = "blac
  labs(x = "Total Unique Quote Tokens",
       y = "Frequency") +
  facet_wrap(~Final_Setting, ncol = 1, strip.position = "left") +
  scale_fill_manual(values = palette, name = "Total Rounds") +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +
  theme_bw(base_size = 16) +
  theme(panel.grid.minor.x = element_blank(),
        panel.grid.major.x = element_line(linewidth = 1),
        panel.grid.minor.y = element_line(linewidth = 0.25))
```


```r
colfunc <- colorRampPalette(c(correctColor,"white",incorrectColor))
palette <- colfunc(length(levels(as.factor(debater_turns$quote_length_bin))))
debater_turns %>%
  filter(`Max judge rounds by room` != "0" & !is.na(Final_Setting)) %>%
  group_by(Final_Setting) %>%
  summarise(avg = mean(`Max judge rounds by room`),
            lower_ci = t.test(`Max judge rounds by room`)$conf.int[1],
            upper_ci = t.test(`Max judge rounds by room`)$conf.int[2]) %>%
  ggplot(aes(x = avg)) +
  geom_histogram(data = debater_turns %>% filter(`Max judge rounds by room` != "0" & !is.na(Final_Setti
                 aes(x = `Max judge rounds by room`, fill = quote_length_bin),
                 position='stack',
                 binwidth = 1,
                 color = "black",
                 size = 0.1) +
  geom_vline(aes(xintercept = avg), size=1, color = "black") +
  geom_rect(aes(xmin = lower_ci, xmax = upper_ci, ymin = -Inf, ymax = Inf), alpha = 0.25, fill = "black
  labs(x = "Total Rounds",
       y = "Frequency") +
  facet_wrap(~Final_Setting, ncol = 1, strip.position = "left", scales = "free_y") +
  scale_fill_manual(values = palette, name = "Total\nUnique\nTokens*") +
  scale_x_continuous(breaks = 1:25, expand = expansion(mult = c(0, 0))) +
  scale_y_continuous(expand = expansion(mult = c(0, 0.05))) +
  theme_bw(base_size = 16) +
  theme(panel.grid.minor.x = element_blank(),
        panel.grid.major.x = element_line(linewidth = 1),
```

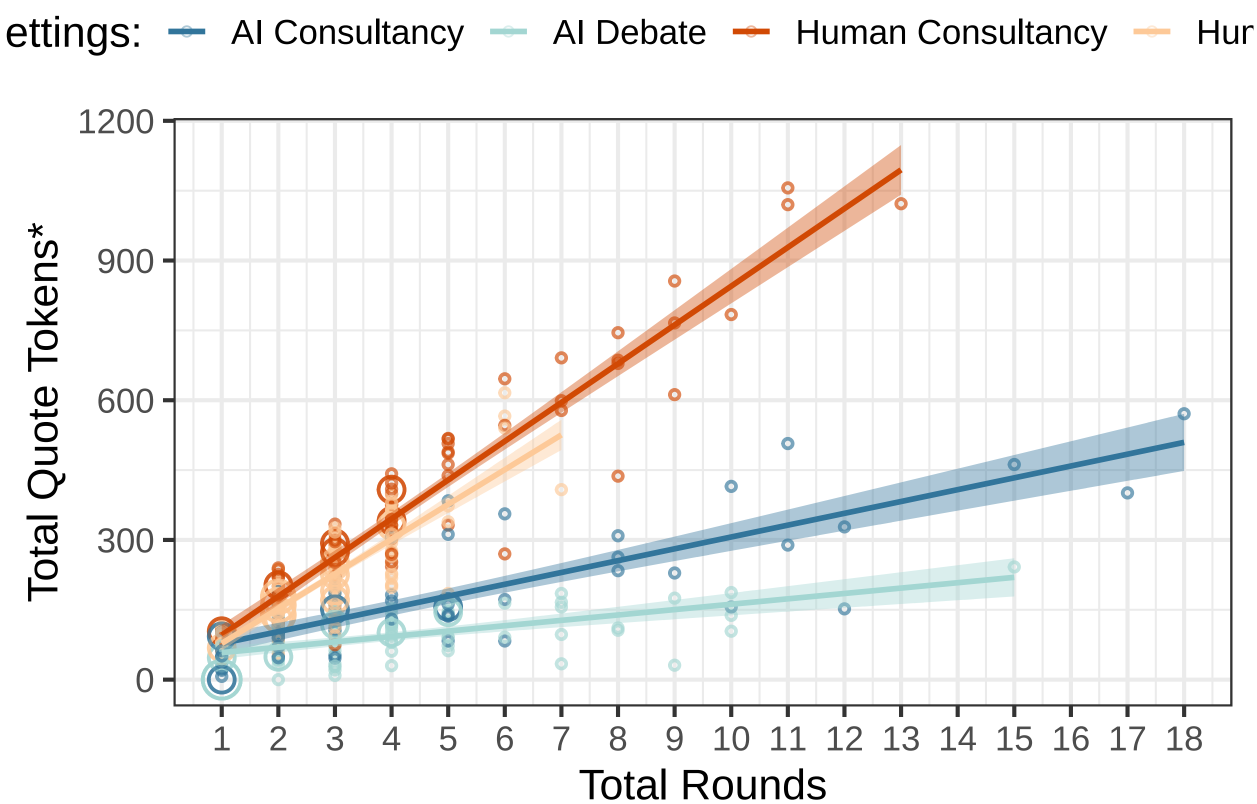
```
       panel.grid.minor.y = element_line(linewidth = 0.25))
```




```
debater_turns %>%
  filter(`Max judge rounds by room` != "0" & !is.na(Final_Setting)) %>%
  group_by(`Max judge rounds by room`, quote_length, Final_Setting) %>%
  mutate(counts = n()) %>%
  ggplot() +
  geom_point(aes(x = `Max judge rounds by room`,
                 y = quote_length,
                 color = Final_Setting,
                 #fill = Final_Setting,
                 size = counts,
                 stroke = 1),
             alpha = 0.65,
             shape = 21) +
  geom_smooth(aes(x = `Max judge rounds by room`,
                  y = quote_length,
                  color = Final_Setting,
                  fill = Final_Setting),  # Added fill aesthetic here
              method = "lm",
              linetype = "solid") +
  labs(x = "Total Rounds",
       y = "Total Quote Tokens*",
       color = "Settings:") +
  guides(size = "none", fill = "none",
```

```
        color = guide_legend(override.aes = list(fill = "white"))) +
  scale_x_continuous(breaks = 0:25) +
  scale_color_manual(values = c("#32759b", "#a3d6d2", "#d14904", "#fdc998")) +
  scale_fill_manual(values = c("#32759b", "#a3d6d2", "#d14904", "#fdc998")) +  # Set fill colors here
  theme_bw(base_size = 16) +
  theme(legend.position = "top")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
ggsave("efficiency_rounds_tokens.png", plot = last_plot(), width = 13, height = 8, bg = "white", dpi = 3
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

```
debater_turns %>%
  filter(`Max judge rounds by room` != "0" & !is.na(Final_Setting)) %>%
  group_by(Final_Setting) %>%
  summarise(avg = mean(`Max judge rounds by room`),
            lower_ci = t.test(`Max judge rounds by room`)$conf.int[1],
            upper_ci = t.test(`Max judge rounds by room`)$conf.int[2])
```

```
## # A tibble: 4 x 4
##   Final_Setting       avg lower_ci upper_ci
##   <chr>             <dbl>    <dbl>    <dbl>
```

```
## 1 AI Consultancy     4.18     3.31     5.06
## 2 AI Debate          3.82     3.26     4.37
## 3 Human Consultancy  4.02     3.53     4.52
## 4 Human Debate       2.75     2.57     2.93
```

```
debater_turns %>%
  filter(`Max judge rounds by room` & !is.na(Final_Setting)) %>%
  group_by(Final_Setting) %>%
  summarise(avg = mean(quote_length),
            lower_ci = t.test(quote_length)$conf.int[1],
            upper_ci = t.test(quote_length)$conf.int[2])
```

```
## # A tibble: 4 x 4
##   Final_Setting       avg lower_ci upper_ci
##   <chr>             <dbl>    <dbl>    <dbl>
## 1 AI Consultancy    159.     131.      186.
## 2 AI Debate          90.5     79.4     102.
## 3 Human Consultancy 347.     304.      391.
## 4 Human Debate      208.     192.      224.
```

Length of debates, stratified

```
all_turns = turns.merge(
        judgments_online[["Room name", "Honest debater", "Dishonest debater", "Question", "Article ID",
                 "Speed annotator accuracy","Untimed annotator context","Untimed annotator context bins
        how="left",
        on="Room name",
    )
```

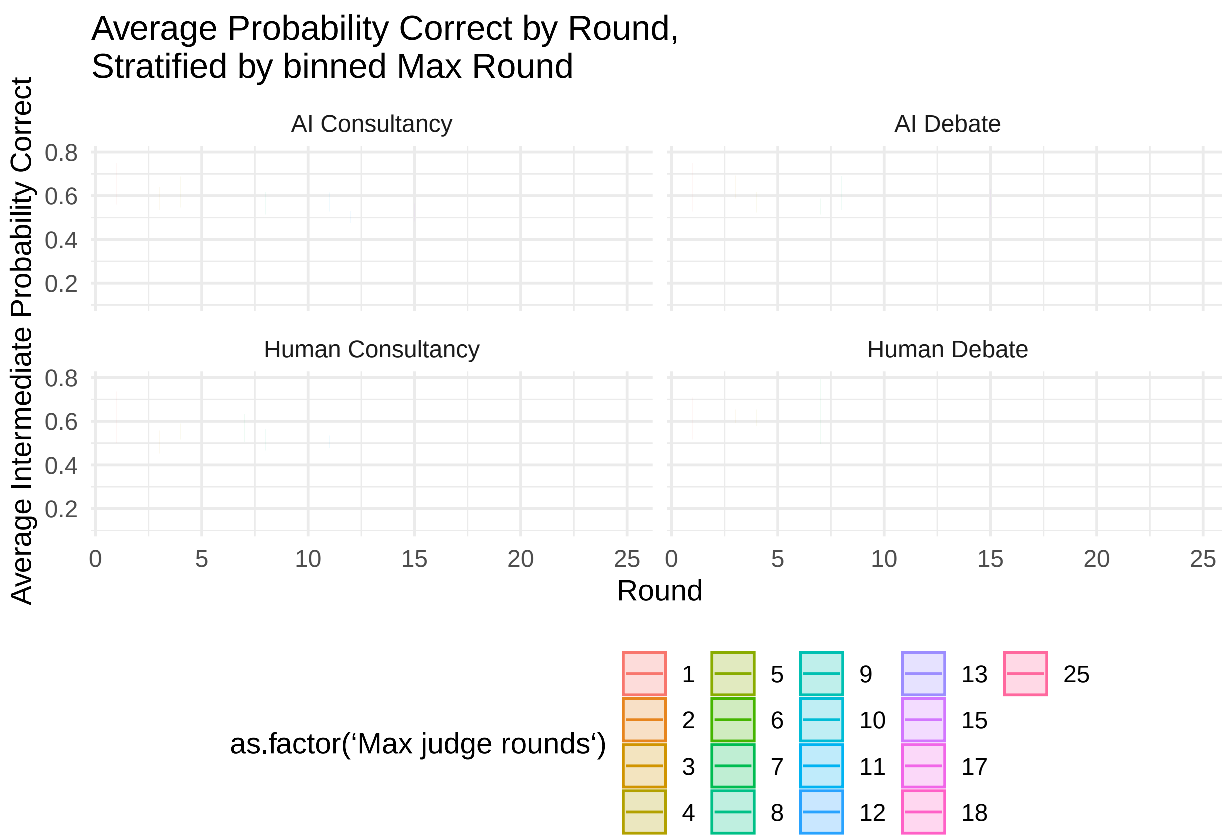
```
strat <- py$all_turns
strat <- subset(strat, strat$Role == "Judge")
strat <- strat %>%
  group_by(`Room name`, Participant) %>%
  mutate(`Max judge rounds` = max(`Number of judge continues`, na.rm = TRUE)) %>%
  ungroup()
```

```
## Warning: There were 114 warnings in ‘mutate()‘.
## The first warning was:
## i In argument: ‘Max judge rounds = max(‘Number of judge continues‘, na.rm =
##   TRUE)‘.
## i In group 3: ‘Room name = "a-pail-of-air-4"‘, ‘Participant = "Jackson Petty"‘.
## Caused by warning in ‘max()‘:
## ! no non-missing arguments to max; returning -Inf
## i Run ‘dplyr::last_dplyr_warnings()‘ to see the 113 remaining warnings.
```

```
# Bootstrap mean function
bootstrap_mean <- function(data, indices) {
  return(mean(data[indices], na.rm = TRUE))
}
```

```r
# Extract unique bin values
unique_bins <- levels(strat$`Max judge rounds`)[2:length(levels(strat$`Max judge rounds`))]

# Create a decreasing sequence of alpha values
alpha_values <- seq(1, 0.1, length.out = length(unique_bins))

# Create a named vector for mapping
alpha_map <- setNames(alpha_values, unique_bins)

strat %>%
  filter(`Max judge rounds` != "0"& !is.na(Final_Setting)) %>%
  group_by(Final_Setting, `Number of judge continues`, `Max judge rounds`) %>%
  do({
    boot_result <- boot(data = .$`Probability correct`, statistic = bootstrap_mean, R = 1000)
    data.frame(
      mean_accuracy = mean(boot_result$t, na.rm = TRUE),
      lower_ci = quantile(boot_result$t, 0.025, na.rm = TRUE),
      upper_ci = quantile(boot_result$t, 0.975, na.rm = TRUE)
    )
  }) %>%
  ggplot(aes(x = `Number of judge continues`, y = mean_accuracy, col = as.factor(`Max judge rounds`)))
  geom_ribbon(aes(ymin = lower_ci, ymax = upper_ci, fill = as.factor(`Max judge rounds`), group = as.fa
  labs(title = "Average Probability Correct by Round, \nStratified by binned Max Round",
       x = "Round",
       y = "Average Intermediate Probability Correct") +
  geom_line() +
  #scale_alpha_manual(values = alpha_map) +
  facet_wrap(~Final_Setting) +
  theme_minimal() +
  theme(legend.position = "bottom") +
  guides(alpha = "none")
```

```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```




```r
strat <- strat %>%
  mutate(
    `Max judge rounds bin` = case_when(
      `Max judge rounds` <= 0 ~ "0",
      `Max judge rounds` <= 2 ~ "1-2",
      `Max judge rounds` <= 4 ~ "3-4",
      `Max judge rounds` <= 6 ~ "5-6",
      `Max judge rounds` <= 8 ~ "7-8",
      TRUE ~ "9+"
    )
  ) %>%
  mutate(
    `Max judge rounds bin` = factor(
      `Max judge rounds bin`,
      levels = rev(c("0","1-2","3-4","5-6", "7-8","9+")),
      ordered = TRUE
    )
  )

table(strat$`Max judge rounds`)
```

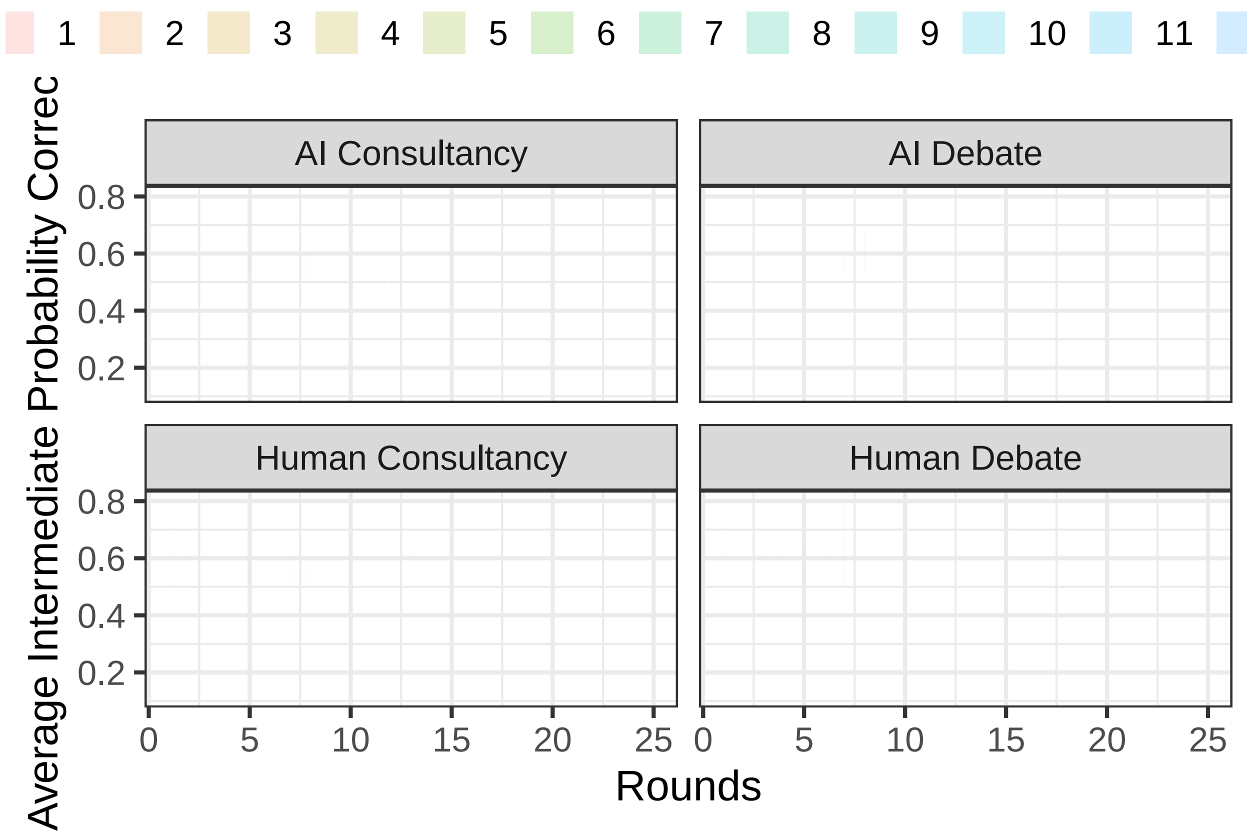
```
##
## -Inf    1    2    3    4    5    6    7    8    9   10   11   12   13   15   17
##  328  122  294  440  385  192  119   72   90   60   99   48   26   14   32   18
##   18   25
##   19   26
```

```r
table(strat$`Max judge rounds bin`)
```

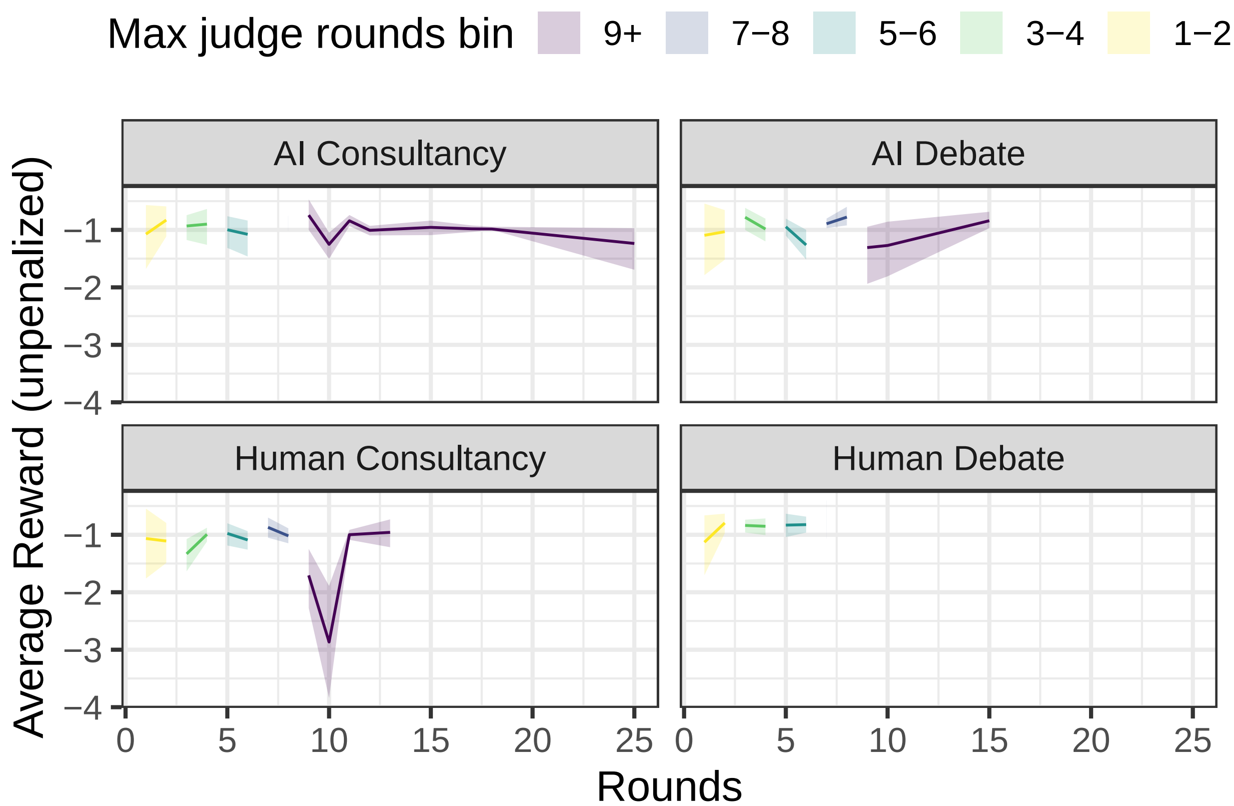
```
##
##  9+ 7-8 5-6 3-4 1-2   0
## 342 162 311 825 416 328
```

```r
strat %>%
  filter(`Max judge rounds` != "0" & !is.na(Final_Setting)) %>%  # Remove entries with "0" bin
  group_by(Final_Setting, `Number of judge continues`, `Max judge rounds`) %>%
  do({
    boot_result <- boot(data = .$`Probability correct`, statistic = bootstrap_mean, R = 1000)
    data.frame(
      mean_accuracy = mean(boot_result$t, na.rm = TRUE),
      lower_ci = quantile(boot_result$t, 0.025, na.rm = TRUE),
      upper_ci = quantile(boot_result$t, 0.975, na.rm = TRUE)
    )
  }) %>%
  ggplot(aes(x = `Number of judge continues`, y = mean_accuracy, col = as.factor(`Max judge rounds`))) +
  geom_ribbon(aes(ymin = lower_ci, ymax = upper_ci, fill = as.factor(`Max judge rounds`), color = NULL)
  labs(x = "Rounds",
       y = "Average Intermediate Probability Correct",
       col = "Settings") +  # Rename the legend
  geom_line() +
  facet_wrap(~Final_Setting) +
  guides(alpha = "none", color = "none", fill = guide_legend(nrow = 1)) +  # Specify that legend should
  theme_bw(base_size = 16) +
  theme(legend.position = "top")  # Specify legend position
```

```
## ‘geom_line()‘: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## ‘geom_line()‘: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## ‘geom_line()‘: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
## ‘geom_line()‘: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```

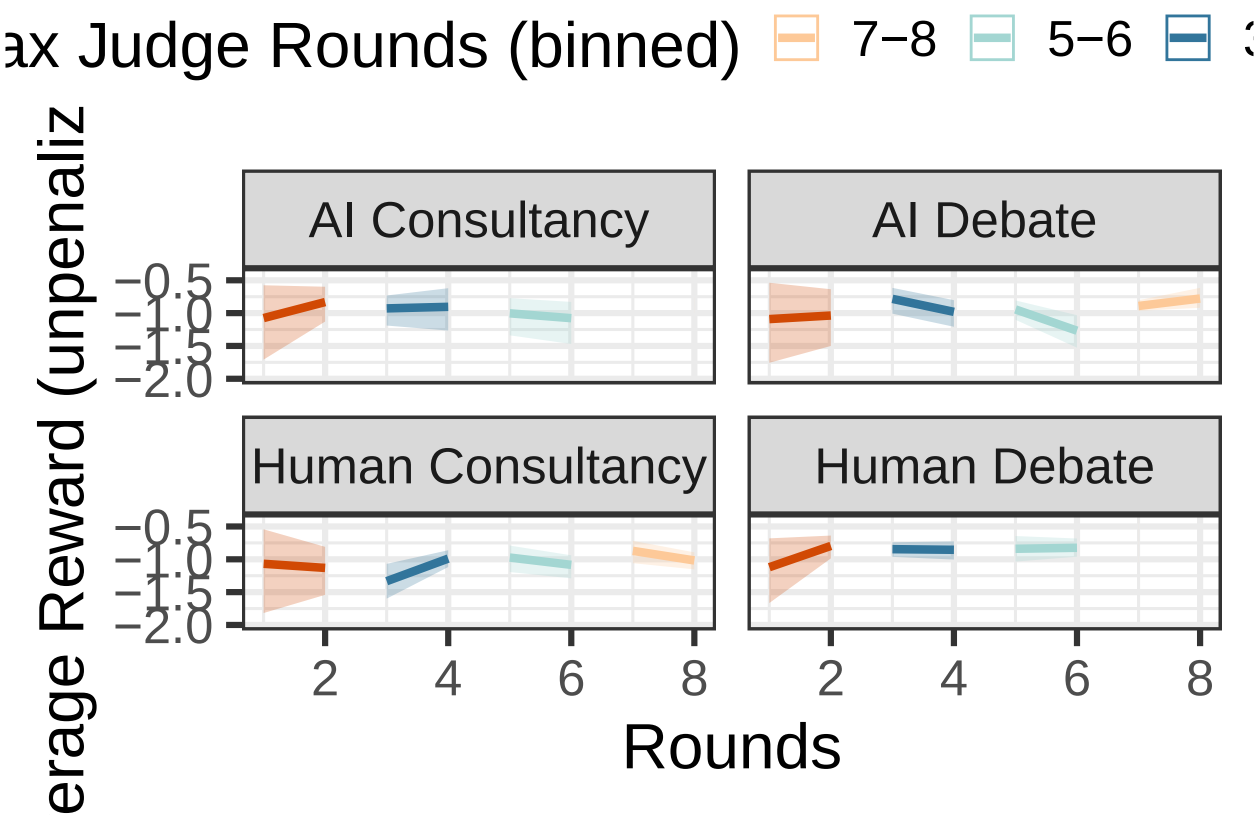



```r
strat$reward_unpenalized <- log2(strat$`Probability correct`)
strat %>%
  filter(`Max judge rounds` != "0" & !is.na(Final_Setting)) %>%  # Remove entries with "0" bin
  group_by(Final_Setting, `Number of judge continues`, `Max judge rounds bin`) %>%
  do({
    boot_result <- boot(data = .$reward_unpenalized, statistic = bootstrap_mean, R = 1000)
    data.frame(
      mean_accuracy = mean(boot_result$t, na.rm = TRUE),
      lower_ci = quantile(boot_result$t, 0.025, na.rm = TRUE),
      upper_ci = quantile(boot_result$t, 0.975, na.rm = TRUE)
    )
  }) %>%
  ggplot(aes(x = `Number of judge continues`, y = mean_accuracy, col = `Max judge rounds bin`)) +
  geom_ribbon(aes(ymin = lower_ci, ymax = upper_ci, fill = `Max judge rounds bin`, color = NULL), alpha
  labs(x = "Rounds",
       y = "Average Reward (unpenalized)",
       col = "Settings") +  # Rename the legend
  geom_line() +
  facet_wrap(~Final_Setting) +
  guides(alpha = "none", color = "none", fill = guide_legend(nrow = 1)) +  # Specify that legend should
  theme_bw(base_size = 16) +
  theme(legend.position = "top")  # Specify legend position
```




```
colfunc <- colorRampPalette(c("grey", "black"), bias = 3)
palette <- colfunc(length(c("0-1", "2-3", "4-5", "6-7")))

strat %>%
  filter(`Max judge rounds` != "0" & !is.na(Final_Setting) &
         `Max judge rounds bin` %in% c("1-2","3-4","5-6", "7-8")) %>%  # Remove entries with "9+" bin
  group_by(Final_Setting, `Number of judge continues`, `Max judge rounds bin`) %>%
  do({
    boot_result <- boot(data = .$reward_unpenalized, statistic = bootstrap_mean, R = 1000)
    data.frame(
      mean_accuracy = mean(boot_result$t, na.rm = TRUE),
      lower_ci = quantile(boot_result$t, 0.025, na.rm = TRUE),
      upper_ci = quantile(boot_result$t, 0.975, na.rm = TRUE)
    )
  }) %>%
  ggplot(aes(x = `Number of judge continues`, y = mean_accuracy, col = `Max judge rounds bin`)) +
  #geom_hline(yintercept = -1, linetype="solid", color = "black") +
  geom_ribbon(aes(ymin = lower_ci, ymax = upper_ci, fill = `Max judge rounds bin`, color = NULL), alpha
  labs(x = "Rounds",
       y = "Average Reward (unpenalized)",
       col = "Settings") +  # Rename the legend
  geom_line(linewidth=1.5) +
  facet_wrap(~Final_Setting) +
  #scale_color_brewer(palette = "Set1") +
  #scale_fill_brewer(palette = "Set1") +
  scale_color_manual(values = c("#fdc998", "#a3d6d2","#32759b", "#d14904")) +
```

```
scale_fill_manual(values = c( "#fdc998", "#a3d6d2", "#32759b","#d14904")) +
guides(alpha = "none", fill = "none", color = guide_legend(nrow = 1, title = "Max Judge Rounds (binned
theme_bw(base_size = 24) +
theme(legend.position = "top") +
coord_cartesian(ylim = c(-2, NA))
```




```
ggsave("main_info_accuracy.png", plot = last_plot(), width = 13, height = 8, bg = "white", dpi = 300)


# Split strat into subsets based on Max judge rounds
strat_split <- split(strat, strat$`Max judge rounds bin`)

# Define the analysis function
analysis_function <- function(df) {
  df %>%
    filter(`Max judge rounds` != "0" & !is.na(Final_Setting)) %>%
    group_by(Final_Setting, `Number of judge continues`) %>%
    summarise(mean_prob_correct = mean(log2(`Probability correct`), na.rm = TRUE)) %>%
    mutate(diff = mean_prob_correct - lag(mean_prob_correct)) %>%
    summarise(mean_diff = mean(diff, na.rm=TRUE))
}

# Apply the analysis function to each subset
results <- map(strat_split, analysis_function)
```

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

```
# Get the unique values of 'Max judge rounds' (assuming they are in the same order as 'results')
max_judge_rounds_values <- as.numeric(names(results))
```

```
## Warning: NAs introduced by coercion
```

```
# Add a column to each data frame in 'results' to identify the value of 'Max judge rounds'
results <- map2(results, max_judge_rounds_values, ~ mutate(.x, `Max judge rounds` = .y))

# Combine the list of data frames into a single data frame
final_result <- bind_rows(results)
```

Time (offline judging..?)

```
# Convert to datetime
judgments["Offline judging start time"] = pd.to_datetime(judgments["Offline judging start time"], unit=
judgments["Offline judging end time"] = pd.to_datetime(judgments["Offline judging end time"], unit="ms"

# Calculate offline judging time in minutes
judgments["Offline judging time"] = (judgments["Offline judging end time"] - judgments["Offline judging


print(f"Number of offline judgments on consultancies:\n{judgments[judgments['Setting'].str.contains('Co
```

```
## Number of offline judgments on consultancies:
## count      15.000000
## mean      388.789360
## std      1155.486869
## min         1.169167
## 25%         2.330417
## 50%         6.138050
## 75%        12.285925
## max      4369.697933
## Name: Offline judging time, dtype: float64
## Only 13...
```

```
# Filter out rows with NaT values
valid_judging_time = judgments["Offline judging time"].dropna()
```

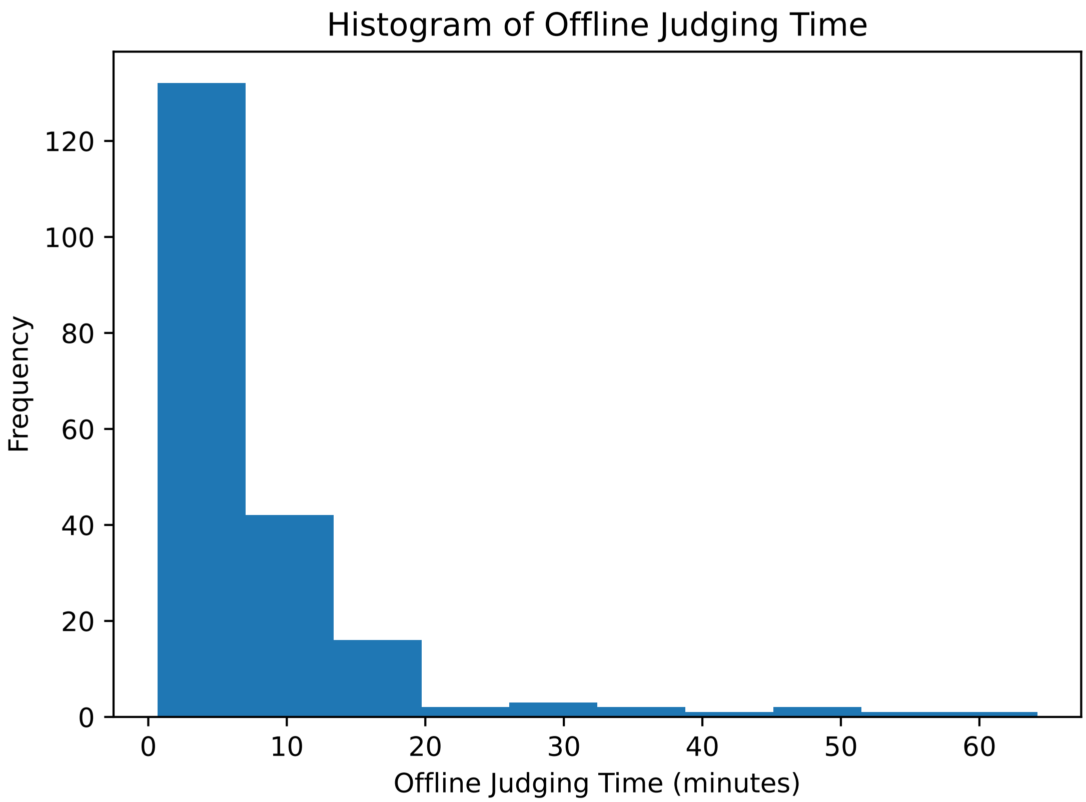
```python
# Calculate summary statistics
summary_stats = valid_judging_time.describe()
print(summary_stats)
```

```
## count      212.000000
## mean       245.256583
## std       1343.567769
## min          0.667467
## 25%          2.865396
## 50%          5.260225
## 75%         10.241529
## max      14202.493917
## Name: Offline judging time, dtype: float64
```

```python
# Filter judgments with offline judging time above 65 minutes
filtered_judgments = judgments[(judgments["Offline judging time"] < 65) & (judgments["Untimed annotator

# Print filtered judgments
# print("Filtered judgments with offline judging time above 65 minutes:")
print(filtered_judgments['Offline judging time'].describe())
```

```
## count    202.000000
## mean       7.961557
## std        9.269747
## min        0.667467
## 25%        2.845525
## 50%        5.124833
## 75%        8.636038
## max       64.173267
## Name: Offline judging time, dtype: float64
```

```python
# Create the histogram
plt.hist(filtered_judgments['Offline judging time'], bins=10)

# Set labels and title
plt.xlabel("Offline Judging Time (minutes)")
plt.ylabel("Frequency")
plt.title("Histogram of Offline Judging Time")

# Display the histogram
plt.show()
```




```
aggregates = {
    'Final probability correct': 'mean',
    'Untimed annotator context': 'mean'
}
filtered_judgments = filtered_judgments.groupby('Offline judging time').agg(aggregates).reset_index()
```

Analysis

Question Difficulty

confounder rounds, quotes

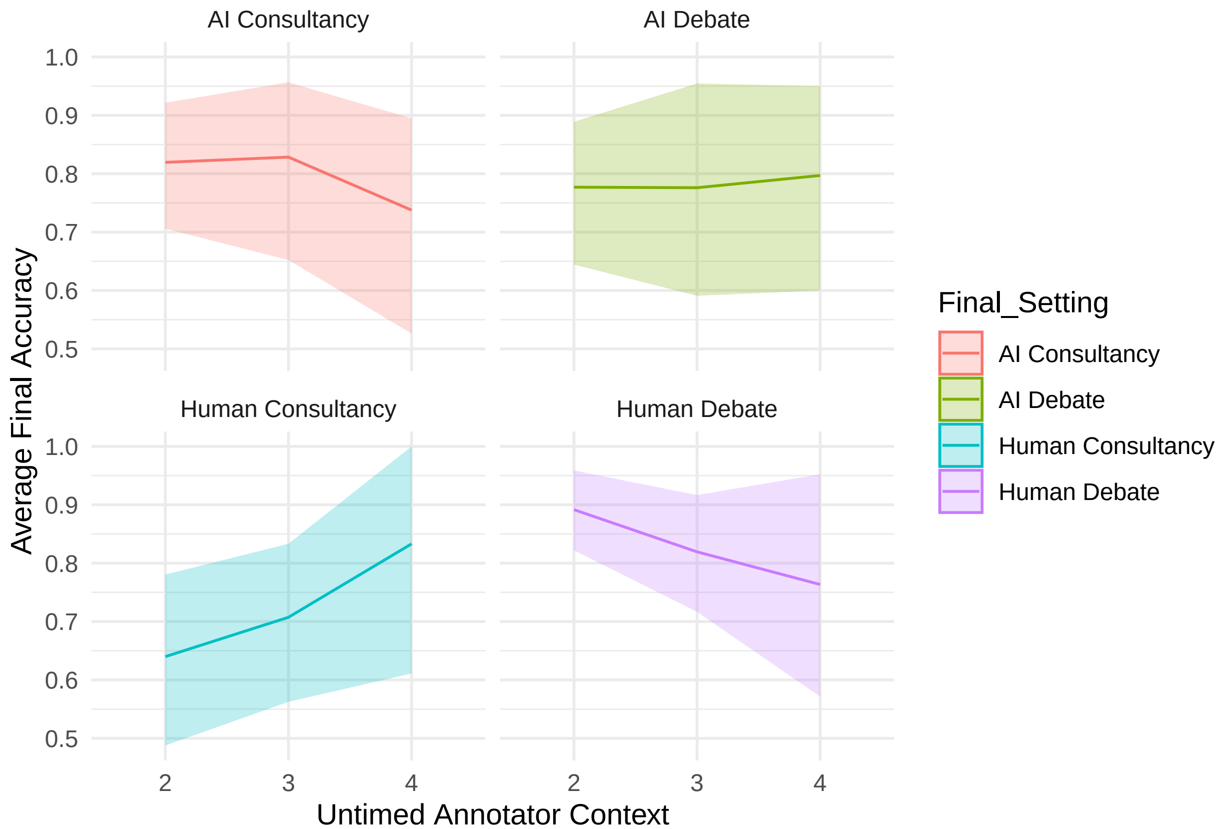
```
judgments["Number of judge continues bins"] = pd.cut(
    judgments["Number of judge continues"],
    bins=[0, 3, 6, 9, float('inf')],  # bin edges
    labels=['1-3', '4-6', '7-9', '10+'],  # labels for the resulting bins
    right=True  # includes the right edge of the bin
)
aggregated_df = judgments.groupby(["Setting", "Number of judge continues bins"])["Final_Accuracy"].agg(
    Proportion_True=lambda x: x.mean(),
    Total_Count="size"
).reset_index()
pd.set_option('display.max_columns', None)
print(aggregated_df)
```

```
##                          Setting Number of judge continues bins  \
## 0      AI Consultancy Dishonest                            1-3
## 1      AI Consultancy Dishonest                            4-6
## 2      AI Consultancy Dishonest                            7-9
## 3      AI Consultancy Dishonest                            10+
## 4         AI Consultancy Honest                            1-3
## 5         AI Consultancy Honest                            4-6
## 6         AI Consultancy Honest                            7-9
## 7         AI Consultancy Honest                            10+
## 8                     AI Debate                            1-3
## 9                     AI Debate                            4-6
## 10                    AI Debate                            7-9
## 11                    AI Debate                            10+
## 12  Human Consultancy Dishonest                            1-3
## 13  Human Consultancy Dishonest                            4-6
## 14  Human Consultancy Dishonest                            7-9
## 15  Human Consultancy Dishonest                            10+
## 16     Human Consultancy Honest                            1-3
## 17     Human Consultancy Honest                            4-6
## 18     Human Consultancy Honest                            7-9
## 19     Human Consultancy Honest                            10+
## 20                 Human Debate                            1-3
## 21                 Human Debate                            4-6
## 22                 Human Debate                            7-9
## 23                 Human Debate                            10+
##
##     Proportion_True  Total_Count
## 0          0.962963           27
## 1          0.833333            6
## 2          1.000000            2
## 3          0.400000            5
## 4          0.740741           27
## 5          0.777778           18
## 6          1.000000            3
## 7          0.625000            8
## 8          0.843137           51
## 9          0.740741           27
## 10         0.700000           10
## 11         0.500000            4
## 12         0.483871           31
## 13         0.633333           30
## 14         0.833333            6
## 15         0.500000            2
## 16         0.928571           28
## 17         0.833333           18
## 18         0.833333            6
## 19         0.500000            2
## 20         0.870370          324
## 21         0.862069           58
## 22         1.000000            1
## 23              NaN            0
```

```python
pd.reset_option('display.max_columns')

total_counts_for_setting = judgments.groupby('Final_Setting').size()
result = judgments.groupby(["Final_Setting", "Untimed annotator context bins", "Number of judge continu
    Proportion_True=pd.NamedAgg(column='Final_Accuracy', aggfunc=lambda x: x.mean()),
    Context_Count=pd.NamedAgg(column='Final_Accuracy', aggfunc='size'),
    Proportion_Context=pd.NamedAgg(column='Final_Setting', aggfunc=lambda x: len(x) / total_counts_for_s
).reset_index()
print(f'Is it number of rounds (meaning more evidence) that confounds the consultancy accuracy?:\n{resul
```

```
## Is it number of rounds (meaning more evidence) that confounds the consultancy accuracy?:
##      Final_Setting  ... Proportion_Context
## 0   AI Consultancy  ...                NaN
## 1   AI Consultancy  ...           0.010417
## 2   AI Consultancy  ...                NaN
## 3   AI Consultancy  ...                NaN
## 4   AI Consultancy  ...           0.291667
## ..             ...  ...                ...
## 59    Human Debate  ...                NaN
## 60    Human Debate  ...           0.078329
## 61    Human Debate  ...           0.018277
## 62    Human Debate  ...                NaN
## 63    Human Debate  ...                NaN
##
## [64 rows x 6 columns]
```

```r
judgments$`Untimed annotator context bins` <- as.factor(judgments$`Untimed annotator context bins`)

bootstrap_mean <- function(data, indices) {
  return(mean(data[indices], na.rm = TRUE))
}

judgments_online %>%
  group_by(`Untimed annotator context bins`, Final_Setting) %>%
  do({
    boot_result <- boot(data = .$Final_Accuracy, statistic = bootstrap_mean, R = 1000)
    data.frame(
      mean_accuracy = mean(boot_result$t, na.rm = TRUE),
      lower_ci = quantile(boot_result$t, 0.025),
      upper_ci = quantile(boot_result$t, 0.975)
    )
  }) %>%
  ggplot(aes(x = `Untimed annotator context bins`, y = mean_accuracy, color = Final_Setting, group = Fi
  geom_line() +
  geom_ribbon(aes(ymin = lower_ci, ymax = upper_ci, fill = Final_Setting, color = NULL), alpha = 0.25) +
  labs(y = "Average Final Accuracy", x = "Untimed Annotator Context") +
  theme_minimal() +
  facet_wrap(~ Final_Setting)
```



Judge Skill

Judge "Experience"

```
test_trend <- judgments_online %>%
  arrange(Final_Setting, Participant, `End time`) %>%
  group_by(Final_Setting, `End time`) %>%
  summarise(Final_Accuracy=as.numeric(Final_Accuracy)) %>%
  spread(key = Final_Setting, value = Final_Accuracy) %>%
  select(-`End time`)
```

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

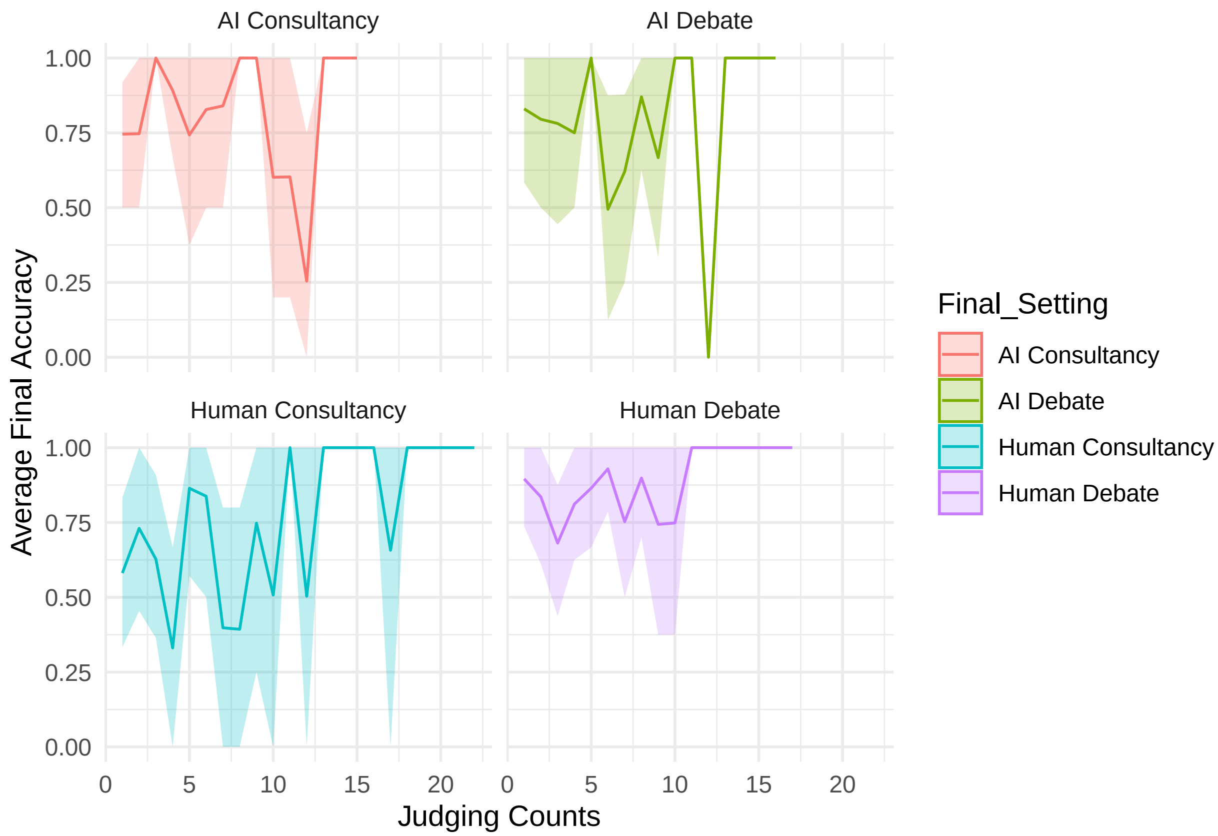
```
test_trend <- judgments_online %>%
  arrange(Final_Setting, Participant, `End time`) %>%
  group_by(Final_Setting, `End time`) %>%
  summarise(Final_Accuracy=as.numeric(Final_Accuracy)) %>%
  spread(key = Final_Setting, value = Final_Accuracy) %>%
  select(-`End time`)
```

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

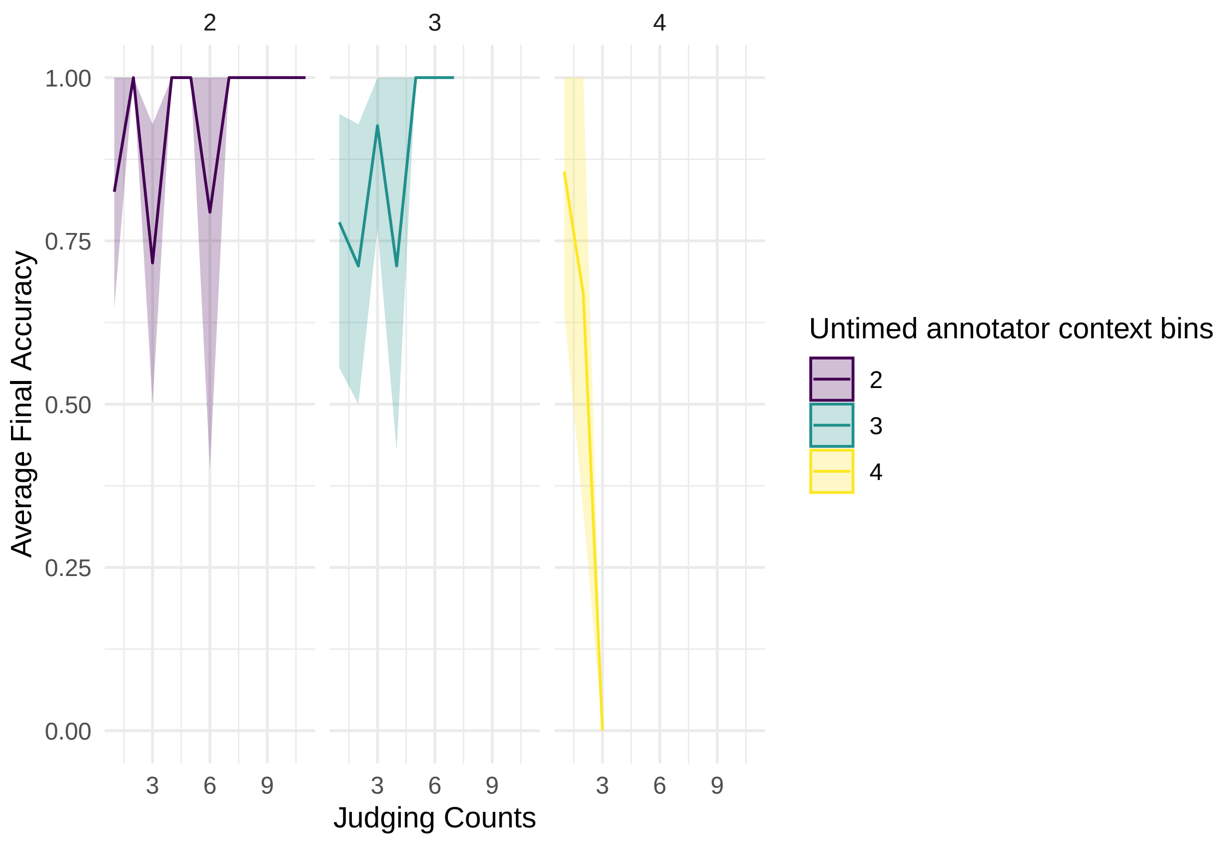
```
library(funtimes)
apply(test_trend, 2, function(x) notrend_test(na.omit(x))$p.value)
```

```
##    AI Consultancy          AI Debate Human Consultancy      Human Debate
##             0.244              0.467             0.133             0.777
```

```
judgments_online %>%
  group_by(Final_Setting, Participant) %>%
  arrange(`End time`) %>%
  mutate(count=row_number()) %>%
  group_by(Final_Setting, count) %>%
  do({
    boot_result <- boot(data = .$Final_Accuracy, statistic = bootstrap_mean, R = 1000)
    data.frame(
      mean_accuracy = mean(boot_result$t, na.rm = TRUE),
      lower_ci = quantile(boot_result$t, 0.025, na.rm = TRUE),
      upper_ci = quantile(boot_result$t, 0.975, na.rm = TRUE)
    )
  }) %>%
  ggplot(aes(x = count, y = mean_accuracy, color = Final_Setting, group = Final_Setting)) +
  geom_line() +
  geom_ribbon(aes(ymin = lower_ci, ymax = upper_ci, fill = Final_Setting, color = NULL), alpha = 0.25) +
  labs(y = "Average Final Accuracy", x = "Judging Counts") +
  theme_minimal() +
  facet_wrap(~ Final_Setting)
```



```
subset(judgments_online, judgments_online['Setting'] == 'Human Debate') %>%
  group_by(`Untimed annotator context bins`, Participant) %>%
  arrange(`End time`) %>%
  mutate(count=row_number()) %>%
  group_by(`Untimed annotator context bins`, count) %>%
  do({
    boot_result <- boot(data = .$Final_Accuracy, statistic = bootstrap_mean, R = 1000)
    data.frame(
      mean_accuracy = mean(boot_result$t, na.rm = TRUE),
      lower_ci = quantile(boot_result$t, 0.025, na.rm = TRUE),
      upper_ci = quantile(boot_result$t, 0.975, na.rm = TRUE)
    )
  }) %>%
  ggplot(aes(x = count, y = mean_accuracy, color = `Untimed annotator context bins`, group = `Untimed an
  geom_line() +
  geom_ribbon(aes(ymin = lower_ci, ymax = upper_ci, fill = `Untimed annotator context bins`, color = NUL
  labs(y = "Average Final Accuracy", x = "Judging Counts") +
  theme_minimal() +
  facet_wrap(~ `Untimed annotator context bins`)
```



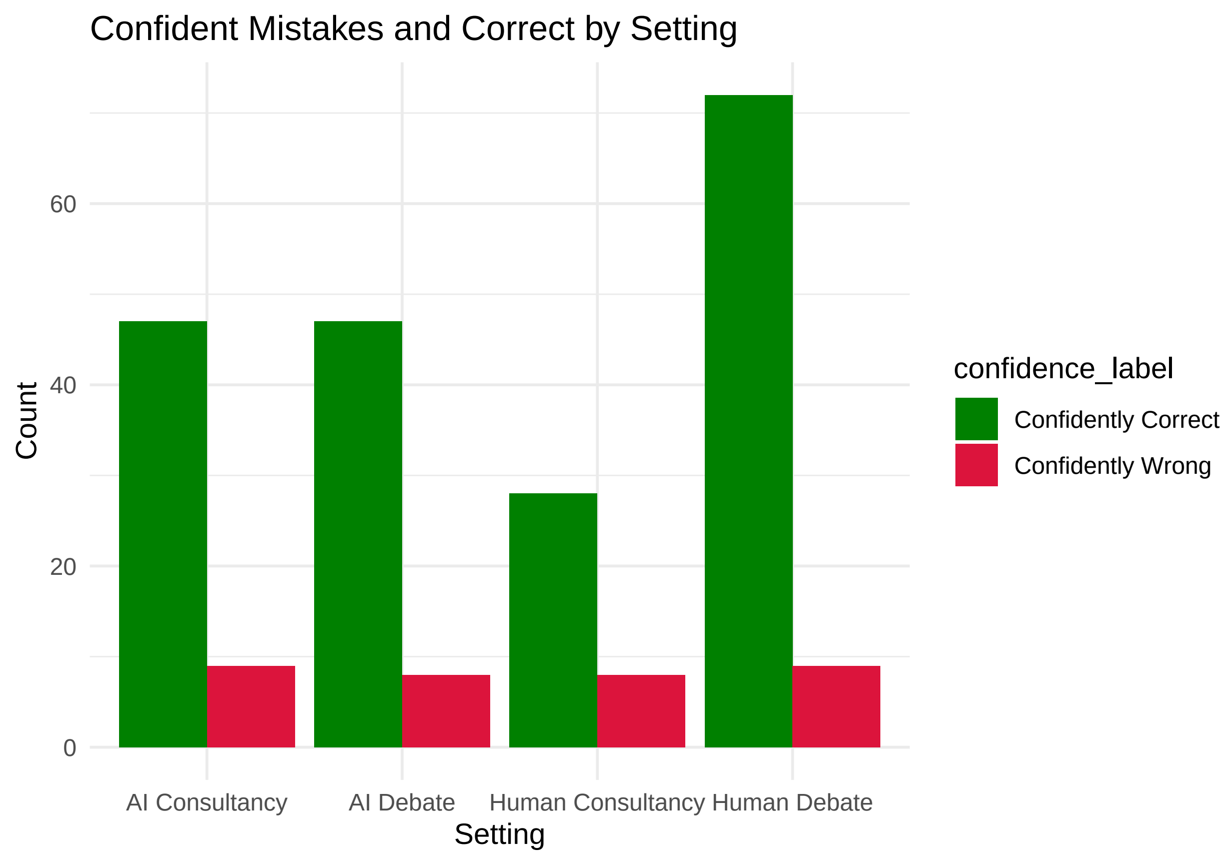
Calibration

S: (1) debaters didnt learn calibration -> calibration over time? S: (2) dishonest debater tricks

```r
library(ggplot2)
library(dplyr)

# Segregate confidently correct and confidently wrong
judgments_online$confidence_label <- case_when(
  judgments_online$`Final probability correct` > 0.95 ~ "Confidently Correct",
  judgments_online$`Final probability correct` < 0.05 ~ "Confidently Wrong",
  TRUE ~ "Neutral"
)

# Filter out only the rows with confidently correct and confidently wrong labels
filtered_data <- judgments_online %>%
  filter(confidence_label != "Neutral")

# Count the occurrences for each setting and confidence label
count_data <- filtered_data %>%
  group_by(`Final_Setting`, confidence_label) %>%
  summarise(count = n())
```

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

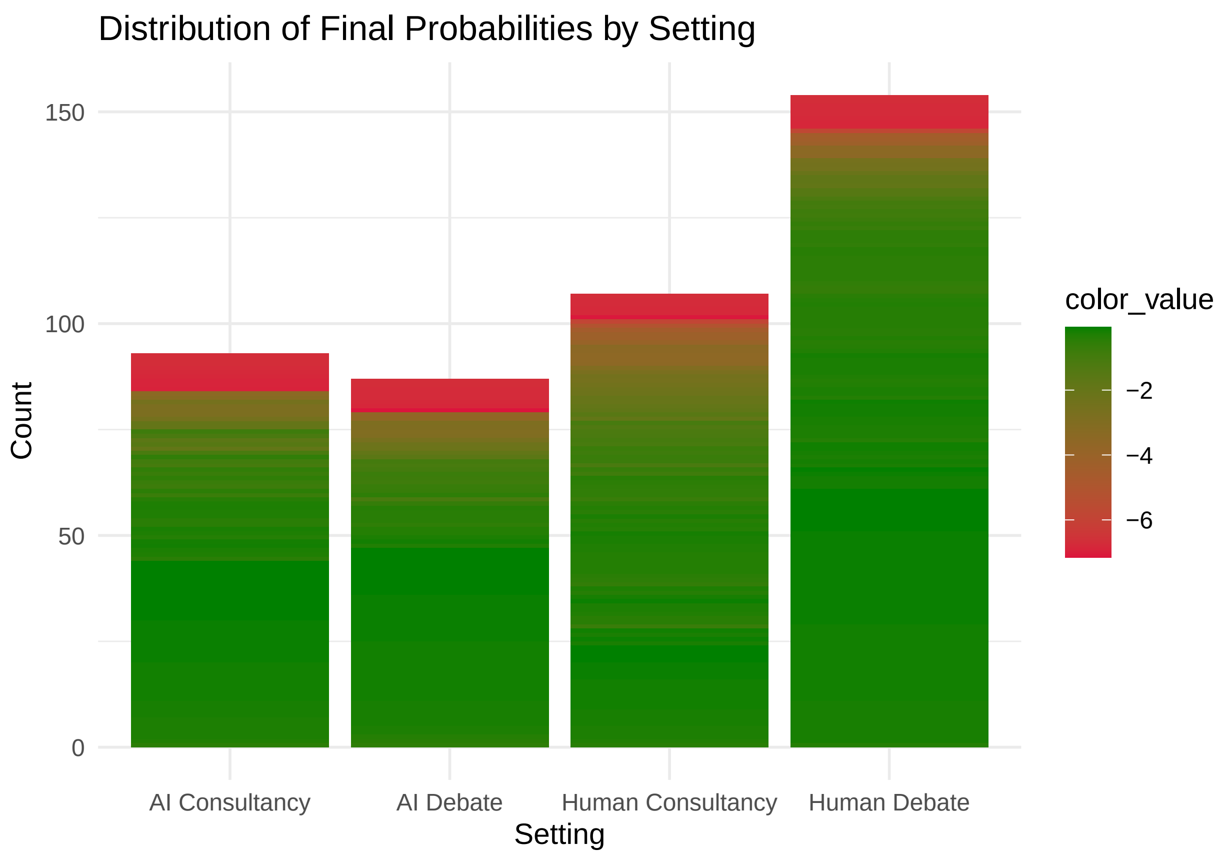
```r
# Plot
ggplot(count_data, aes(x = `Final_Setting`, y = count, fill = confidence_label)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual(values = c("Confidently Correct" = correctColor, "Confidently Wrong" = incorrectCol
  labs(title = "Confident Mistakes and Correct by Setting", y = "Count", x = "Setting") +
  theme_minimal()
```




```
# Calculate the color value for each row
judgments_online$color_value <- log2(judgments_online$`Final probability correct`) - (0.05 * judgments_o

# Count the occurrences for each setting and 'Final probability correct' value
count_data <- judgments_online %>%
  group_by(`Final_Setting`, `Final probability correct`, color_value) %>%
  summarise(count = n())
```

```
## 'summarise()' has grouped output by 'Final_Setting', 'Final probability
## correct'. You can override using the '.groups' argument.
```

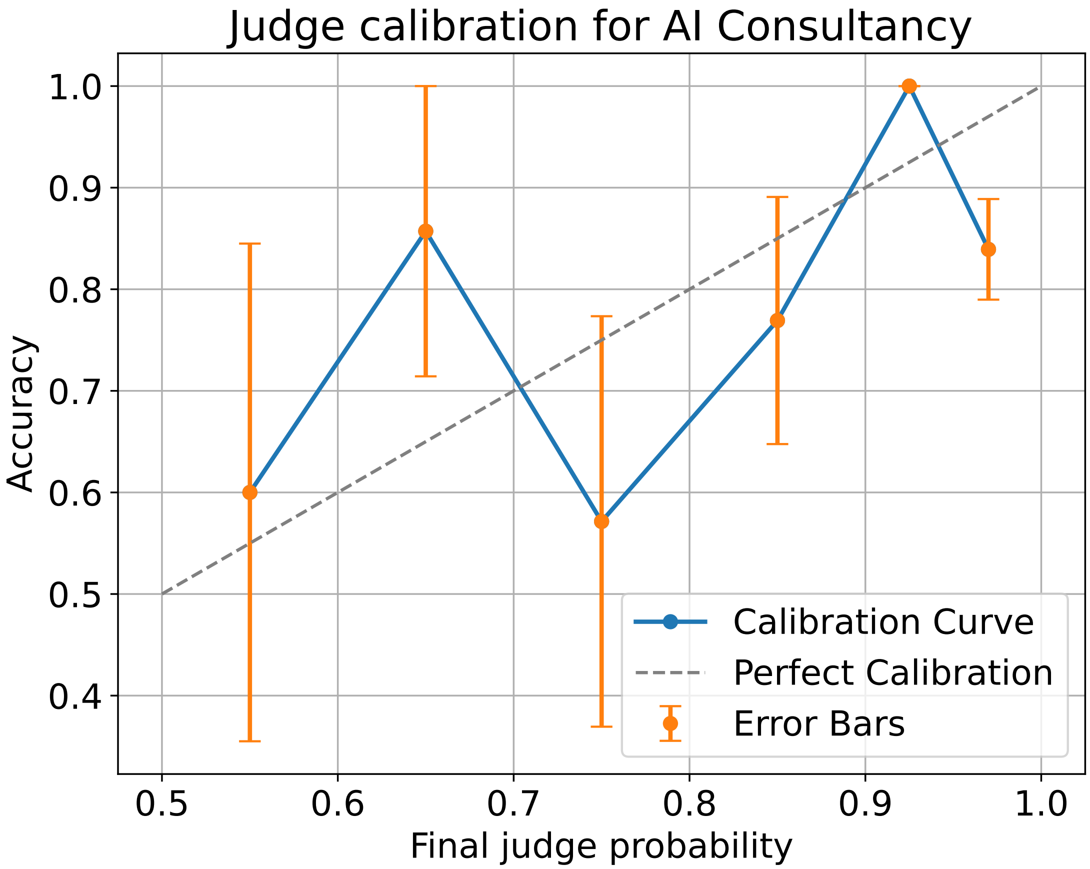
```
# Plot
ggplot(count_data, aes(x = `Final_Setting`, y = count, fill = color_value, group = `Final probability co
  geom_bar(stat = "identity", position = "stack") +
  scale_fill_gradient(low = "#DC143C", high = "#008000") +  # Adjust as needed
  labs(title = "Distribution of Final Probabilities by Setting", y = "Count", x = "Setting") +
  theme_minimal()
```

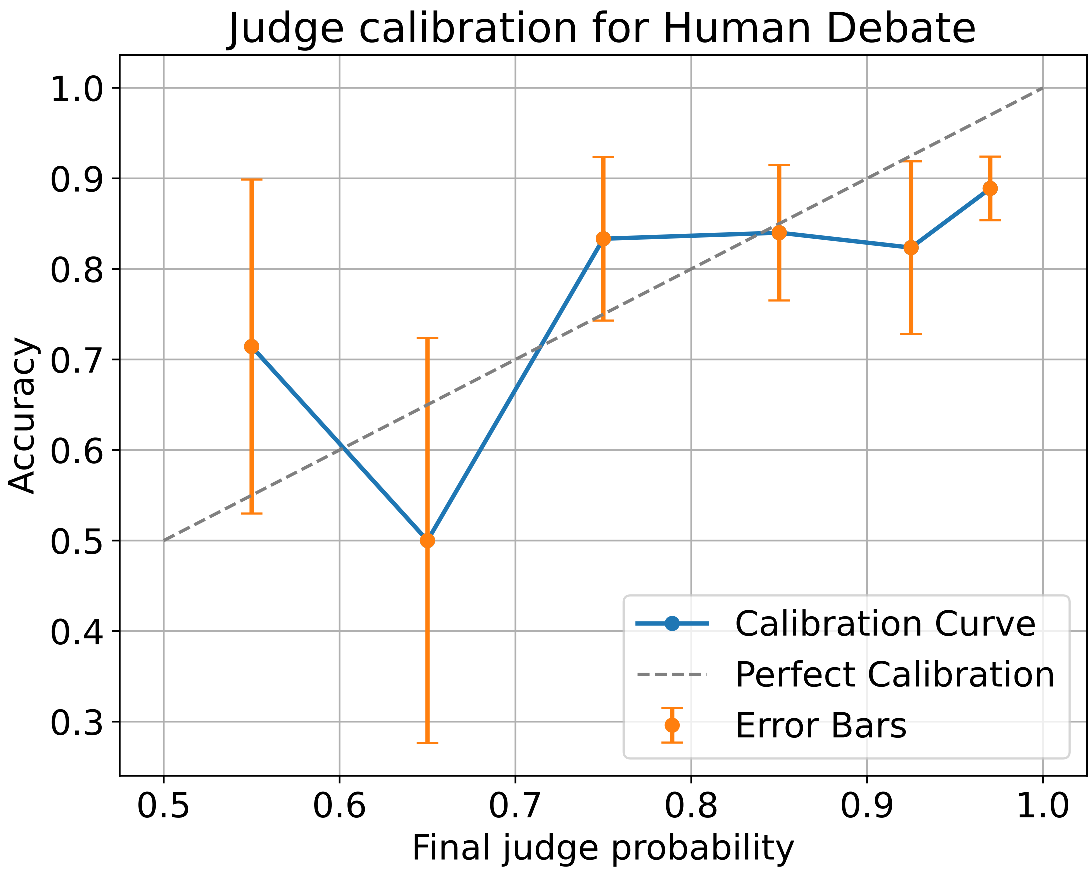


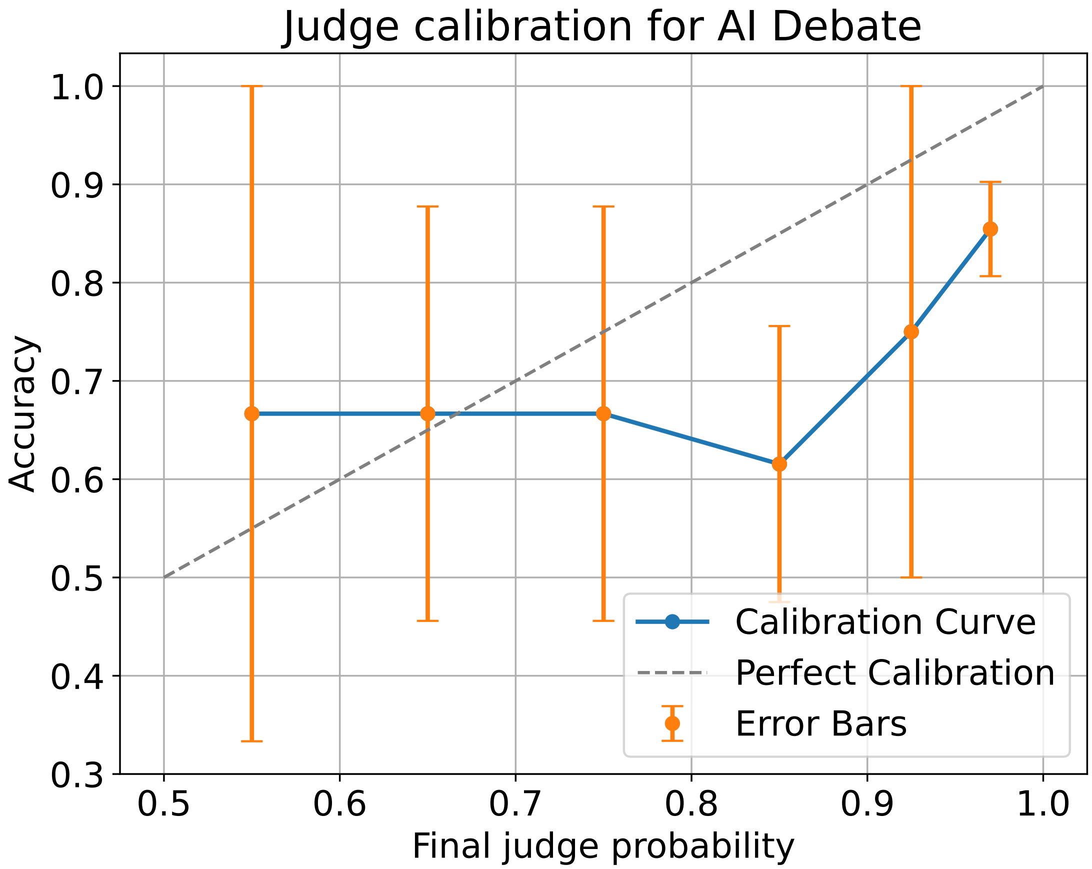

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.calibration import calibration_curve

def calibration_plot(df, setting_name, ax=None):
    df['outcome'] = pd.Series(df['Final probability correct'] > 0.5, dtype=int)
    df['confidence'] = df['Final probability correct'].apply(lambda x: x if x > 0.5 else 1 - x)
    df['bins'] = pd.cut(df['confidence'], [0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99])
    # Group by bins and calculate the mean outcome
    df_grouped = df.groupby('bins')['outcome'].mean().reset_index()
    # Compute standard error in each bin
    std_error = df.groupby('bins')['outcome'].apply(lambda x: x.std() / np.sqrt(len(x)) if len(x) > 1 e
    df_grouped['std_error'] = df['bins'].cat.categories.map(std_error)
    if ax is None:
        plt.rcParams.update({'font.size': 16})
        fig, ax = plt.subplots(figsize=(8, 6))
    # Plot the calibration curve with error bars
    ax.plot(df_grouped['bins'].apply(lambda x: x.mid), df_grouped['outcome'], marker='o', linewidth=2, l
    ax.errorbar(df_grouped['bins'].apply(lambda x: x.mid), df_grouped['outcome'], yerr=df_grouped['std_e
    ax.set_xlabel('Final judge probability')
    ax.set_ylabel('Accuracy')
    ax.set_title(f'Judge calibration for {setting_name}')
    ax.plot([0.5, 1], [0.5, 1], linestyle='--', color='gray', label='Perfect Calibration')
    ax.grid(True)
    ax.legend()
```

```python
    # Calculate ECE
    actual_labels = df['outcome'].values
    predicted_probs = df['Final probability correct'].values
    prob_true, prob_pred = calibration_curve(actual_labels, predicted_probs, n_bins=10)
    ece = np.mean(np.abs(prob_pred - prob_true) * (prob_true.size / len(actual_labels)))
    # Print ECE
    print(f"Expected Calibration Error (ECE) for {setting_name}: {ece:.4f}")
    plt.show()
    plt.rcParams.update({'font.size': plt.rcParamsDefault['font.size']})

# Loop through each unique setting and create a calibration plot
for setting in judgments_online['Final_Setting'].unique():
    setting_df = judgments_online[judgments['Final_Setting'] == setting].copy()
    calibration_plot(setting_df, setting)
```

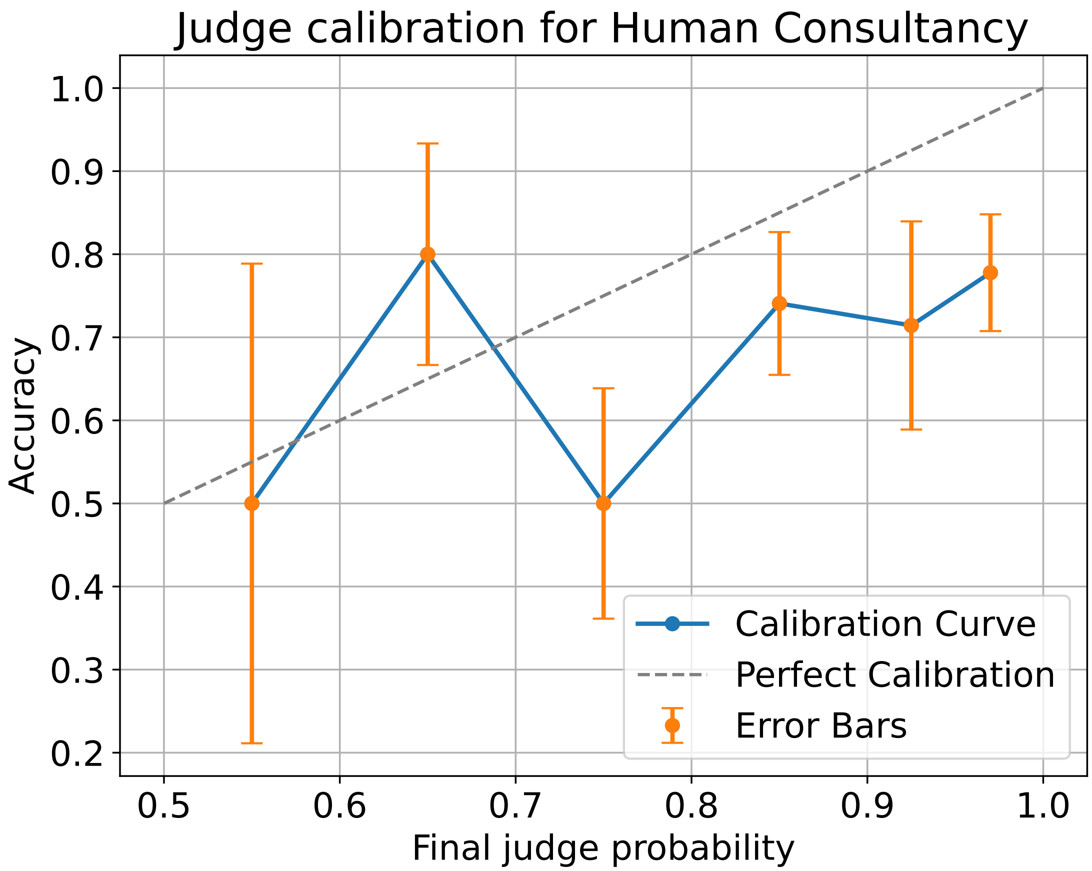
```
## Expected Calibration Error (ECE) for AI Consultancy: 0.0213
## Expected Calibration Error (ECE) for Human Debate: 0.0151
## Expected Calibration Error (ECE) for AI Debate: 0.0268
## Expected Calibration Error (ECE) for Human Consultancy: 0.0220
##
## <string>:4: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
## <string>:4: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
## <string>:4: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
## <string>:4: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
```


Judge calibration for AI Consultancy
Accuracy
Final judge probability
0.4
0.5
0.6
0.7
0.8
0.9
1.0
0.5
0.6
0.7
0.8
0.9
1.0
Calibration Curve
Perfect Calibration
Error Bars


Judge calibration for Human Debate
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.5
0.6
0.7
0.8
0.9
1.0
Final judge probability
Calibration Curve
Perfect Calibration
Error Bars



Judge calibration for AI Debate
Accuracy
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.5
0.6
0.7
0.8
0.9
1.0
Final judge probability
Calibration Curve
Perfect Calibration
Error Bars



Judge Involvement

Judge Mistakes

Debater Skill

```
leaderboard = sessions.merge(
        debates[
            [
                "Room name",
                "Honest debater",
                "Dishonest debater"
            ]
        ],
        how="left",
        on="Room name",
    )



leaderboard = sessions.merge(
```

```
        debates[
            [
                "Room name",
                "Honest debater",
                "Dishonest debater"
            ]
        ],
        how="left",
        on="Room name",
    )
# filter to sessions corresponding to a judge
leaderboard = leaderboard[leaderboard['Final probability correct'].notna()]
from math import log
leaderboard['Judge reward'] = leaderboard.apply(
    lambda row: log(row['Final probability correct']) - (0.05 * row['Number of judge continues']),
    axis=1
)
leaderboard['Honest debater reward'] = leaderboard.apply(
    lambda row: log(row['Final probability correct']),
    axis=1
)
leaderboard['Dishonest debater reward'] = leaderboard.apply(
    lambda row: log(1 - row['Final probability correct']),
    axis=1
)
```

```
leaderboard<-py$leaderboard
leaderboard <- leaderboard %>%
  group_by(Participant) %>%
  mutate(Avg_Judge_Reward = mean(`Judge reward`, na.rm = TRUE)) %>%
  ungroup()
leaderboard <- leaderboard %>%
  group_by(`Honest debater`) %>%
  mutate(Avg_Honest_Debater_Reward = mean(`Honest debater reward`, na.rm = TRUE)) %>%
  ungroup()
leaderboard <- leaderboard %>%
  group_by(`Dishonest debater`) %>%
  mutate(Avg_Dishonest_Debater_Reward = mean(`Dishonest debater reward`, na.rm = TRUE)) %>%
  ungroup()
leaderboard <- leaderboard %>%
  mutate(Rank_Judge = rank(desc(Avg_Judge_Reward)),
         Rank_Honest = rank(desc(Avg_Honest_Debater_Reward)),
         Rank_Dishonest = rank(desc(Avg_Dishonest_Debater_Reward)))
```

```
random.intercept.model = lmer(`Final probability correct` ~  (1|Final_Setting),
                              data = judgments, REML = TRUE)

judgments$random.intercept.preds = predict(random.intercept.model)

colnames(judgments)
```

```
##  [1] "Participant"
##  [2] "base_room_name"
```

```
##  [3] "Room name"
##  [4] "Room start time"
##  [5] "Role"
##  [6] "Is turn"
##  [7] "Is over"
##  [8] "Number of judge continues"
##  [9] "Final probability correct"
## [10] "Offline judging start time"
## [11] "Offline judging end time"
## [12] "other"
## [13] "factual informativeness (comparative).1"
## [14] "factual informativeness (comparative).2"
## [15] "facts versus semantics (single)"
## [16] "factual accuracy (single)"
## [17] "clarity.1"
## [18] "clarity.2"
## [19] "factual accuracy.1"
## [20] "factual accuracy.2"
## [21] "judge reasoning"
## [22] "reason for outcome"
## [23] "protocol"
## [24] "evidence use.1"
## [25] "evidence use.2"
## [26] "evidence in story.1"
## [27] "evidence in story.2"
## [28] "other factors"
## [29] "judge adaptation (single)"
## [30] "evidence in debate.1"
## [31] "evidence in debate.2"
## [32] "interface"
## [33] "evidence in debate (single)"
## [34] "facts versus semantics.1"
## [35] "facts versus semantics.2"
## [36] "clash.1"
## [37] "clash.2"
## [38] "identity guesses.Judge"
## [39] "identity guesses.Debater A"
## [40] "identity guesses.Debater B"
## [41] "judge adaptation.1"
## [42] "judge adaptation.2"
## [43] "subjective correctness"
## [44] "evidence use (single)"
## [45] "factual informativeness (total)"
## [46] "judge strategies"
## [47] "clarity (single)"
## [48] "Debater A"
## [49] "Debater B"
## [50] "Honest debater"
## [51] "Dishonest debater"
## [52] "Is single debater"
## [53] "Has honest debater"
## [54] "Final_Setting"
## [55] "Setting"
## [56] "Question"
```

```
## [57] "Article ID"
## [58] "Story length"
## [59] "Speed annotator accuracy bins"
## [60] "Untimed annotator context bins"
## [61] "Speed annotator accuracy"
## [62] "Untimed annotator context"
## [63] "Is offline"
## [64] "End time"
## [65] "Last modified time"
## [66] "Final_Accuracy"
## [67] "random.intercept.preds"
```

```r
dishonest <- judgments[!is.na(judgments$`Dishonest debater`), ]
model3 <- glm(Final_Accuracy ~ relevel(factor(`Dishonest debater`), 'Shlomo Kofman') + relevel(factor(F
summary(model3)
```

```
##
## Call:
## glm(formula = Final_Accuracy ~ relevel(factor(‘Dishonest debater‘),
##     "Shlomo Kofman") + relevel(factor(Final_Setting), "Human Debate"),
##     family = "binomial", data = judgments[!is.na(judgments$‘Dishonest debater‘),
##         ])
##
## Coefficients: (1 not defined because of singularities)
##                                                                    Estimate
## (Intercept)                                                          0.5712
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Adelle Fernando      0.9663
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Aliyaah Toussaint    2.4639
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Anuj Jain            1.5119
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")David Rein           1.3747
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Emmanuel Makinde    16.9948
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Ethan Rosen          1.4098
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")GPT-4                0.7097
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Jackson Petty        1.6312
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Jessica Li           0.5108
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Julian Michael       2.4245
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Julien Dirani        1.5082
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Max Layden          16.9948
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Noor Mirza-Rashid   -0.1012
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Reeya Kansra         1.4208
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Salsabila Mahdi      1.4965
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Sam Jin              1.3030
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Sean Wang            1.5781
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Shreeram Modi        1.3474
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Vishakh Padmakumar  16.9948
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy              0.6650
## relevel(factor(Final_Setting), "Human Debate")AI Debate                       NA
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy          -1.4147
##                                                                    Std. Error
## (Intercept)                                                            0.6652
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Adelle Fernando        0.7411
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Aliyaah Toussaint      1.2376
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")Anuj Jain              0.8508
## relevel(factor(‘Dishonest debater‘), "Shlomo Kofman")David Rein             0.9074
```

```
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Emmanuel Makinde    2797.4420
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Ethan Rosen            0.8526
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")GPT-4                  0.7116
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Jackson Petty          0.9021
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Jessica Li             0.7455
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Julian Michael         1.2217
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Julien Dirani          1.2520
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Max Layden          3956.1804
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Noor Mirza-Rashid      0.8761
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Reeya Kansra           0.9116
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Salsabila Mahdi        0.7946
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Sam Jin                0.9425
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Sean Wang              0.7718
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Shreeram Modi          0.7509
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Vishakh Padmakumar   863.3096
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy                0.5408
## relevel(factor(Final_Setting), "Human Debate")AI Debate                         NA
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy             0.3230
##                                                                        z value
## (Intercept)                                                              0.859
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Adelle Fernando     1.304
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Aliyaah Toussaint   1.991
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Anuj Jain           1.777
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")David Rein          1.515
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Emmanuel Makinde    0.006
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Ethan Rosen         1.653
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")GPT-4               0.997
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Jackson Petty       1.808
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Jessica Li          0.685
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Julian Michael      1.985
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Julien Dirani       1.205
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Max Layden          0.004
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Noor Mirza-Rashid  -0.116
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Reeya Kansra        1.559
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Salsabila Mahdi     1.883
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Sam Jin             1.382
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Sean Wang           2.045
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Shreeram Modi       1.794
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Vishakh Padmakumar  0.020
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy             1.230
## relevel(factor(Final_Setting), "Human Debate")AI Debate                     NA
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy         -4.380
##                                                                        Pr(>|z|)
## (Intercept)                                                              0.3905
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Adelle Fernando     0.1923
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Aliyaah Toussaint   0.0465
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Anuj Jain           0.0756
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")David Rein          0.1298
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Emmanuel Makinde    0.9952
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Ethan Rosen         0.0982
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")GPT-4               0.3186
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Jackson Petty       0.0706
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Jessica Li          0.4932
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Julian Michael      0.0472
```

```
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Julien Dirani         0.2283
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Max Layden            0.9966
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Noor Mirza-Rashid     0.9080
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Reeya Kansra          0.1191
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Salsabila Mahdi       0.0596
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Sam Jin               0.1668
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Sean Wang             0.0409
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Shreeram Modi         0.0728
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Vishakh Padmakumar    0.9843
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy               0.2188
## relevel(factor(Final_Setting), "Human Debate")AI Debate                        NA
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy         0.0000118
##
## (Intercept)
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Adelle Fernando
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Aliyaah Toussaint  *
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Anuj Jain          .
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")David Rein
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Emmanuel Makinde
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Ethan Rosen        .
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")GPT-4
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Jackson Petty      .
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Jessica Li
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Julian Michael     *
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Julien Dirani
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Max Layden
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Noor Mirza-Rashid
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Reeya Kansra
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Salsabila Mahdi    .
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Sam Jin
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Sean Wang          *
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Shreeram Modi      .
## relevel(factor(`Dishonest debater`), "Shlomo Kofman")Vishakh Padmakumar
## relevel(factor(Final_Setting), "Human Debate")AI Consultancy
## relevel(factor(Final_Setting), "Human Debate")AI Debate
## relevel(factor(Final_Setting), "Human Debate")Human Consultancy         ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 547.18  on 583  degrees of freedom
## Residual deviance: 491.84  on 562  degrees of freedom
## AIC: 535.84
##
## Number of Fisher Scoring iterations: 16
```

```r
result <- judgments_online %>%
  group_by(`Dishonest debater`) %>%
  summarize(
    Win_Rate = sum(Final_Accuracy == "FALSE") / n()
  ) %>%
  ungroup() %>%
  arrange(desc(Win_Rate))
```

```r
result
```

```
## # A tibble: 20 x 2
##    'Dishonest debater' Win_Rate
##    <chr>                  <dbl>
##  1 Shlomo Kofman         0.545
##  2 Salsabila Mahdi       0.357
##  3 Jessica Li            0.353
##  4 Noor Mirza-Rashid     0.333
##  5 Adelle Fernando       0.296
##  6 Reeya Kansra          0.273
##  7 Sam Jin               0.25
##  8 Sean Wang             0.25
##  9 Shreeram Modi         0.24
## 10 GPT-4                 0.192
## 11 <NA>                  0.184
## 12 Anuj Jain             0.143
## 13 Julian Michael        0.125
## 14 Aliyaah Toussaint     0.111
## 15 Ethan Rosen           0.0909
## 16 Jackson Petty         0.0769
## 17 David Rein            0
## 18 Julien Dirani         0
## 19 Max Layden            0
## 20 Vishakh Padmakumar     0
```

```r
result1 <- judgments_online %>%
  group_by(`Honest debater`) %>%
  summarize(
    Win_Rate = sum(Final_Accuracy == "TRUE") / n()
  ) %>%
  ungroup() %>%
  arrange(desc(Win_Rate))

result1
```

```
## # A tibble: 20 x 2
##    'Honest debater'   Win_Rate
##    <chr>                 <dbl>
##  1 Julian Michael        1
##  2 Julien Dirani         1
##  3 Noor Mirza-Rashid     1
##  4 Sean Wang             0.96
##  5 Jessica Li            0.923
##  6 Salsabila Mahdi       0.917
##  7 Adelle Fernando       0.905
##  8 Reeya Kansra          0.9
##  9 Vishakh Padmakumar    0.857
## 10 Shlomo Kofman         0.833
## 11 Anuj Jain             0.8
## 12 David Rein            0.8
## 13 Shreeram Modi         0.8
```

```
## 14 Ethan Rosen          0.786
## 15 GPT-4                0.775
## 16 Aliyaah Toussaint    0.714
## 17 <NA>                 0.680
## 18 Jackson Petty        0.667
## 19 Sam Jin              0.667
## 20 Emmanuel Makinde     0
```

```
# Filter for high win rate debaters
high_win_rate_debaters <- result1 %>%
  filter(Win_Rate > 0.90)  # Set the threshold for high win rate

# Filter original data for debates with 'Debate' in Final_Setting
filtered_data <- judgments_online %>%
  filter(grepl("Debate", Final_Setting))

# Find cases where high win rate debaters lost
cases_high_win_rate_lost <- filtered_data %>%
  filter(`Honest debater` %in% high_win_rate_debaters$`Honest debater` & Final_Accuracy != "TRUE")

cases_high_win_rate_lost
```

```
##     Participant          base_room_name                Room name
## 146   Anuj Jain          survival-type-          survival-type-5
## 214 Ethan Rosen the-great-nebraska-sea- the-great-nebraska-sea-0
## 289  Jessica Li                     rx-                     rx-1
##     Room start time  Role Is turn Is over Number of judge continues
## 146   1681159356736 Judge   FALSE    TRUE                         5
## 214   1683321454611 Judge   FALSE    TRUE                         2
## 289   1683298141840 Judge   FALSE    TRUE                         2
##     Final probability correct Offline judging start time
## 146                      0.33                        NaN
## 214                      0.01                        NaN
## 289                      0.01                        NaN
##     Offline judging end time other factual informativeness (comparative).1
## 146                      NaN  <NA>                                       4
## 214                      NaN  <NA>                                       1
## 289                      NaN  <NA>                                       4
##     factual informativeness (comparative).2 facts versus semantics (single)
## 146                                       4                             NaN
## 214                                       1                             NaN
## 289                                       2                             NaN
##     factual accuracy (single) clarity.1 clarity.2 factual accuracy.1
## 146                       NaN         3         3                NaN
## 214                       NaN         2         2                NaN
## 289                       NaN         4         1                NaN
##     factual accuracy.2 judge reasoning
## 146                NaN               3
## 214                NaN               1
## 289                NaN               4
##                                                                                                      reason for out
## 146                                                                                                                i
## 214 I thought "like" was over-technical compared to what these questions typically ask for. I was wr
## 289                              B's last arg was literally 2 sentences, and A's ev was very convinci
```

```
##     protocol evidence use.1 evidence use.2 evidence in story.1
## 146     <NA>            NaN            NaN                 NaN
## 214     <NA>            NaN            NaN                 NaN
## 289     <NA>            NaN            NaN                 NaN
##     evidence in story.2 other factors judge adaptation (single)
## 146                 NaN          <NA>                       NaN
## 214                 NaN          <NA>                       NaN
## 289                 NaN          <NA>                       NaN
##     evidence in debate.1 evidence in debate.2 interface
## 146                    2                    4      <NA>
## 214                    3                    2      <NA>
## 289                    4                    2      <NA>
##     evidence in debate (single) facts versus semantics.1
## 146                         NaN                        3
## 214                         NaN                        1
## 289                         NaN                        1
##     facts versus semantics.2 clash.1 clash.2 identity guesses.Judge
## 146                        3       3       3                   <NA>
## 214                        3       4       4                   <NA>
## 289                        3       4       2                   <NA>
##     identity guesses.Debater A identity guesses.Debater B judge adaptation.1
## 146                       <NA>                       <NA>                  3
## 214                       <NA>                       <NA>                  4
## 289                       <NA>                       <NA>                  4
##     judge adaptation.2 subjective correctness evidence use (single)
## 146                  3                    NaN                   NaN
## 214                  3                    NaN                   NaN
## 289                  2                    NaN                   NaN
##     factual informativeness (total) judge strategies clarity (single)
## 146                               3             <NA>              NaN
## 214                               1             <NA>              NaN
## 289                               3             <NA>              NaN
##           Debater A       Debater B  Honest debater Dishonest debater
## 146 Adelle Fernando     Ethan Rosen Adelle Fernando       Ethan Rosen
## 214 Salsabila Mahdi       Sean Wang       Sean Wang   Salsabila Mahdi
## 289    Reeya Kansra Adelle Fernando Adelle Fernando      Reeya Kansra
##     Is single debater Has honest debater Final_Setting      Setting
## 146             FALSE               TRUE  Human Debate Human Debate
## 214             FALSE               TRUE  Human Debate Human Debate
## 289             FALSE               TRUE  Human Debate Human Debate
##                                                                   Question
## 146 How did the planet of Niobe compare to others that Earth was exploring?
## 214                                           How is this article written?
## 289   What were the specialties of the Red and Green Doctors, respectively?
##     Article ID Story length Speed annotator accuracy bins
## 146      51395        25010                           0.2
## 214      50893        21233                           0.2
## 289      60412        22224                           0.2
##     Untimed annotator context bins Speed annotator accuracy
## 146                              3                0.1666667
## 214                              3                0.2000000
## 289                              3                0.2000000
##     Untimed annotator context Is offline            End time
## 146                  2.750000      FALSE 2023-04-17 17:12:59
```

```
## 214                  3.333333      FALSE 2023-05-08 17:14:28
## 289                  2.800000      FALSE 2023-06-22 15:18:02
##      Last modified time Final_Accuracy Human Consultancy Sample
## 146 2023-04-28 12:29:25          FALSE                     FALSE
## 214 2023-06-22 15:20:58          FALSE                     FALSE
## 289 2023-06-22 15:18:02          FALSE                     FALSE
##     AI Consultancy Sample Human Debate Sample AI Debate Sample Sample
## 146                 FALSE               FALSE            FALSE  FALSE
## 214                 FALSE                TRUE            FALSE   TRUE
## 289                 FALSE                TRUE            FALSE   TRUE
##     Consultancy Sample initial_question_weights
## 146              FALSE                0.5000000
## 214              FALSE                0.2000000
## 289              FALSE                0.3333333
##     initial_question_weights_grouped_setting
## 146                                      0.5
## 214                                      0.5
## 289                                      0.5
##     sampled_consultancies_all_debates_weights
## 146                                 0.5000000
## 214                                 0.2000000
## 289                                 0.3333333
##     sampled_consultancies_all_debates_weights_setting
## 146                                               0.5
## 214                                               0.5
## 289                                               0.5
##     sampled_consultancies_all_debates_weights_grouped_setting
## 146                                                       0.5
## 214                                                       0.5
## 289                                                       0.5
##     sampled_consultancies_debates_weights
## 146                                  0.00
## 214                                  0.25
## 289                                  0.50
##     sampled_consultancies_debates_weights_setting
## 146                                             0
## 214                                             1
## 289                                             1
##     sampled_consultancies_debates_weights_grouped_setting
## 146                                                     0
## 214                                                     1
## 289                                                     1
##                                   check Reward penalty 0.5  fpc
## 146            Anuj Jainsurvival-type-5          -4.099462 0.33
## 214 Ethan Rosenthe-great-nebraska-sea-0          -7.643856 0.01
## 289                      Jessica Lirx-1          -7.643856 0.01
##      confidence_label color_value
## 146           Neutral   -1.849462
## 214 Confidently Wrong   -6.743856
## 289 Confidently Wrong   -6.743856
```

```r
# Filter for high win rate debaters
high_win_rate_debaters <- result %>%
  filter(Win_Rate > 0.20)  # Set the threshold for high win rate
```

```r
# Filter original data for debates with 'Debate' in Final_Setting
filtered_data <- judgments_online %>%
  filter(grepl("Debate", Final_Setting))

# Find cases where high win rate debaters lost
cases_high_win_rate_lost <- filtered_data %>%
  filter(`Dishonest debater` %in% high_win_rate_debaters$`Dishonest debater` & Final_Accuracy != "FALSE

cases_high_win_rate_lost
```

```
##            Participant                      base_room_name
## 21     Adelle Fernando                           monopoly-
## 43     Adelle Fernando                    tollivers-orbit-
## 78   Aliyaah Toussaint                                 rx-
## 81   Aliyaah Toussaint                 stranger-from-space-
## 91   Aliyaah Toussaint        the-long-remembered-thunder-
## 94   Aliyaah Toussaint     the-princess-and-the-physicist-
## 99   Aliyaah Toussaint                 the-starsent-knaves-
## 113           Anuj Jain                        cosmic-yoyo-
## 136           Anuj Jain               out-of-the-iron-womb-
## 140           Anuj Jain                    planet-of-dread-
## 149           Anuj Jain                the-air-of-castor-oil-
## 177          David Rein                           monopoly-
## 179          David Rein            peggy-finds-the-theatre-
## 185          David Rein                 stalemate-in-space-
## 186          David Rein                 stranger-from-space-
## 191          David Rein              the-great-nebraska-sea-
## 202         Ethan Rosen                        cosmic-yoyo-
## 211         Ethan Rosen                 stranger-from-space-
## 215         Ethan Rosen                 the-man-who-was-six-
## 216         Ethan Rosen                   the-monster-maker-
## 219        Jackson Petty atom-mystery-young-atom-detective-
## 236        Jackson Petty                           muck-man-
## 240        Jackson Petty                                 rx-
## 241        Jackson Petty                     silence-isdeadly-
## 254        Jackson Petty     the-princess-and-the-physicist-
## 270          Jessica Li                     doctor-universe-
## 276          Jessica Li                 how-to-make-friends-1
## 290          Jessica Li                     silence-isdeadly-
## 306          Jessica Li     the-princess-and-the-physicist-
## 324      Julian Michael                           monopoly-
## 331      Julian Michael                 stranger-from-space-
## 332      Julian Michael                       survival-type-
## 338      Julian Michael                   the-monster-maker-
## 342      Julian Michael        the-spicy-sound-of-success-
## 348        Julien Dirani                 manners-and-customs-
## 356   Noor Mirza-Rashid                     doctor-universe-
## 366   Noor Mirza-Rashid                              volpla-
## 378        Reeya Kansra                 how-to-make-friends-
## 387        Reeya Kansra                           muck-man-
## 401        Reeya Kansra                   the-monster-maker-
## 411     Salsabila Mahdi                         break-a-leg-
## 414     Salsabila Mahdi                        cosmic-yoyo-
```

```
## 421    Salsabila Mahdi              manners-and-customs-
## 424    Salsabila Mahdi                         muck-man-
## 425    Salsabila Mahdi                  planet-of-dread-
## 429    Salsabila Mahdi                 silence-isdeadly-
## 431    Salsabila Mahdi              stranger-from-space-
## 433    Salsabila Mahdi               the-happy-castaway-
## 436    Salsabila Mahdi             the-reluctant-heroes-
## 439    Salsabila Mahdi              the-starsent-knaves-
## 457            Sam Jin               coming-of-the-gods-
## 519            Sam Jin            venus-is-a-mans-world-
## 542          Sean Wang              lost-in-translation-
## 547          Sean Wang          peggy-finds-the-theatre-
## 553          Sean Wang                    survival-type-
## 559          Sean Wang                     the-cool-war-
## 570          Sean Wang                           volpla-
## 607      Shlomo Kofman             out-of-the-iron-womb-
## 611      Shlomo Kofman               pied-piper-of-mars-
## 615      Shlomo Kofman                               rx-
## 634      Shlomo Kofman                  the-starbusters-
## 645      Shreeram Modi                      cosmic-yoyo-
## 649      Shreeram Modi                    in-the-garden-
## 655      Shreeram Modi          peggy-finds-the-theatre-
## 656      Shreeram Modi         phone-me-in-central-park-
## 666      Shreeram Modi              the-man-who-was-six-
## 685 Vishakh Padmakumar               stalemate-in-space-
## 687 Vishakh Padmakumar            the-air-of-castor-oil-
## 688 Vishakh Padmakumar         the-desert-and-the-stars-
## 691 Vishakh Padmakumar                the-monster-maker-
##                              Room name Room start time  Role Is turn Is over
## 21                          monopoly-1   1680552464768 Judge   FALSE    TRUE
## 43                   tollivers-orbit-1   1681765942714 Judge   FALSE    TRUE
## 78                                rx-3   1683298141840 Judge   FALSE    TRUE
## 81                 stranger-from-space-0   1683298716462 Judge   FALSE    TRUE
## 91         the-long-remembered-thunder-1   1689876270711 Judge   FALSE    TRUE
## 94      the-princess-and-the-physicist-4   1682112300045 Judge   FALSE    TRUE
## 99                 the-starsent-knaves-2   1688757372245 Judge   FALSE    TRUE
## 113                        cosmic-yoyo-0   1681159027164 Judge   FALSE    TRUE
## 136               out-of-the-iron-womb-0   1689876275997 Judge   FALSE    TRUE
## 140                    planet-of-dread-2   1680829456935 Judge   FALSE    TRUE
## 149              the-air-of-castor-oil-5   1680552962919 Judge   FALSE    TRUE
## 177                         monopoly-2   1680552464768 Judge   FALSE    TRUE
## 179            peggy-finds-the-theatre-4   1682110072206 Judge   FALSE    TRUE
## 185                 stalemate-in-space-0   1677532762430 Judge   FALSE    TRUE
## 186                stranger-from-space-4   1683298716462 Judge   FALSE    TRUE
## 191              the-great-nebraska-sea-1   1683321454611 Judge   FALSE    TRUE
## 202                        cosmic-yoyo-3   1681159027164 Judge   FALSE    TRUE
## 211                stranger-from-space-5   1683298716462 Judge   FALSE    TRUE
## 215                the-man-who-was-six-1   1676313105423 Judge   FALSE    TRUE
## 216                  the-monster-maker-4   1681159292566 Judge   FALSE    TRUE
## 219 atom-mystery-young-atom-detective-0   1689949095893 Judge   FALSE    TRUE
## 236                         muck-man-5   1687546720669 Judge   FALSE    TRUE
## 240                               rx-4   1683298141840 Judge   FALSE    TRUE
## 241                  silence-isdeadly-3   1688157095546 Judge   FALSE    TRUE
## 254     the-princess-and-the-physicist-0   1682112300045 Judge   FALSE    TRUE
```

```
## 270                  doctor-universe-0   1680206097221 Judge    FALSE    TRUE
## 276              how-to-make-friends-11  1681724583153 Judge    FALSE    TRUE
## 290                  silence-isdeadly-2  1688157095546 Judge    FALSE    TRUE
## 306    the-princess-and-the-physicist-2  1682112300045 Judge    FALSE    TRUE
## 324                          monopoly-0  1680552464768 Judge    FALSE    TRUE
## 331               stranger-from-space-1  1683298716462 Judge    FALSE    TRUE
## 332                     survival-type-4  1681159356736 Judge    FALSE    TRUE
## 338                 the-monster-maker-3  1681159292566 Judge    FALSE    TRUE
## 342        the-spicy-sound-of-success-4  1679607458871 Judge    FALSE    TRUE
## 348               manners-and-customs-1  1676043334730 Judge    FALSE    TRUE
## 356                   doctor-universe-5  1680206097221 Judge    FALSE    TRUE
## 366                            volpla-2  1680205817615 Judge    FALSE    TRUE
## 378               how-to-make-friends-0  1681724583153 Judge    FALSE    TRUE
## 387                          muck-man-7  1687546765239 Judge    FALSE    TRUE
## 401                 the-monster-maker-1  1681159292566 Judge    FALSE    TRUE
## 411                      break-a-leg-5  1682110823449 Judge    FALSE    TRUE
## 414                       cosmic-yoyo-2  1681159027164 Judge    FALSE    TRUE
## 421               manners-and-customs-0  1676043281654 Judge    FALSE    TRUE
## 424                          muck-man-4  1687546720669 Judge    FALSE    TRUE
## 425                   planet-of-dread-1  1680829456935 Judge    FALSE    TRUE
## 429                  silence-isdeadly-6  1688157095546 Judge    FALSE    TRUE
## 431               stranger-from-space-2  1683298716462 Judge    FALSE    TRUE
## 433                the-happy-castaway-2  1679606564549 Judge    FALSE    TRUE
## 436              the-reluctant-heroes-2  1682965111772 Judge    FALSE    TRUE
## 439               the-starsent-knaves-0  1688757372245 Judge    FALSE    TRUE
## 457                coming-of-the-gods-2  1689020073883 Judge    FALSE    TRUE
## 519             venus-is-a-mans-world-0  1691058680973 Judge    FALSE    TRUE
## 542               lost-in-translation-3  1678404069200 Judge    FALSE    TRUE
## 547           peggy-finds-the-theatre-0  1682090000149 Judge    FALSE    TRUE
## 553                     survival-type-0  1681159356736 Judge    FALSE    TRUE
## 559                      the-cool-war-0  1689949097911 Judge    FALSE    TRUE
## 570                            volpla-3  1680205817615 Judge    FALSE    TRUE
## 607              out-of-the-iron-womb-1  1689876275999 Judge    FALSE    TRUE
## 611                pied-piper-of-mars-8  1689278492513 Judge    FALSE    TRUE
## 615                                rx-5  1683298141840 Judge    FALSE    TRUE
## 634                   the-starbusters-3  1689371609880 Judge    FALSE    TRUE
## 645                       cosmic-yoyo-1  1681159027164 Judge    FALSE    TRUE
## 649                     in-the-garden-6  1680206043370 Judge    FALSE    TRUE
## 655           peggy-finds-the-theatre-2  1682090000149 Judge    FALSE    TRUE
## 656          phone-me-in-central-park-5  1678684819928 Judge    FALSE    TRUE
## 666               the-man-who-was-six-5  1676645924826 Judge    FALSE    TRUE
## 685                stalemate-in-space-2  1677792427135 Judge    FALSE    TRUE
## 687             the-air-of-castor-oil-4  1680552962919 Judge    FALSE    TRUE
## 688          the-desert-and-the-stars-2  1677792315334 Judge    FALSE    TRUE
## 691                 the-monster-maker-5  1681159292566 Judge    FALSE    TRUE
##     Number of judge continues Final probability correct
## 21                          4                      0.70
## 43                          2                      0.90
## 78                          1                      0.99
## 81                          4                      0.99
## 91                          3                      0.98
## 94                          4                      0.99
## 99                          4                      0.85
## 113                         4                      0.99
```

```
## 136                                 4                          0.99
## 140                                 2                          0.99
## 149                                 3                          0.85
## 177                                 3                          0.85
## 179                                 4                          0.90
## 185                                 2                          0.99
## 186                                 4                          0.95
## 191                                 3                          0.95
## 202                                 2                          0.90
## 211                                 2                          0.95
## 215                                 2                          0.80
## 216                                 2                          0.99
## 219                                 6                          0.80
## 236                                 7                          0.99
## 240                                 3                          0.90
## 241                                 3                          0.99
## 254                                 4                          0.95
## 270                                 2                          0.70
## 276                                 2                          0.99
## 290                                 1                          0.99
## 306                                 2                          0.99
## 324                                 3                          0.99
## 331                                 2                          0.99
## 332                                 2                          0.99
## 338                                 3                          0.99
## 342                                 4                          0.99
## 348                                 3                          0.85
## 356                                 4                          0.85
## 366                                 3                          0.95
## 378                                 3                          0.98
## 387                                 4                          0.88
## 401                                 2                          0.96
## 411                                 2                          0.99
## 414                                 2                          0.99
## 421                                 3                          0.99
## 424                                 3                          0.99
## 425                                 3                          0.99
## 429                                 4                          0.99
## 431                                 2                          0.99
## 433                                 3                          0.99
## 436                                 4                          0.99
## 439                                 6                          0.95
## 457                                 3                          0.99
## 519                                 3                          0.99
## 542                                 2                          0.98
## 547                                 2                          0.90
## 553                                 1                          0.98
## 559                                 3                          0.99
## 570                                 2                          0.95
## 607                                 1                          0.94
## 611                                 4                          0.91
## 615                                 4                          0.86
## 634                                 3                          0.97
## 645                                 4                          0.95
```

```
## 649                            2                        0.99
## 655                            1                        0.99
## 656                            2                        0.99
## 666                            3                        0.99
## 685                            3                        0.80
## 687                            2                        0.75
## 688                            3                        0.75
## 691                            5                        0.80
##     Offline judging start time Offline judging end time
## 21                           NaN                      NaN
## 43                           NaN                      NaN
## 78                           NaN                      NaN
## 81                           NaN                      NaN
## 91                           NaN                      NaN
## 94                           NaN                      NaN
## 99                           NaN                      NaN
## 113                          NaN                      NaN
## 136                          NaN                      NaN
## 140                          NaN                      NaN
## 149                          NaN                      NaN
## 177                          NaN                      NaN
## 179                          NaN                      NaN
## 185                          NaN                      NaN
## 186                          NaN                      NaN
## 191                          NaN                      NaN
## 202                          NaN                      NaN
## 211                          NaN                      NaN
## 215                          NaN                      NaN
## 216                          NaN                      NaN
## 219                          NaN                      NaN
## 236                          NaN                      NaN
## 240                          NaN                      NaN
## 241                          NaN                      NaN
## 254                          NaN                      NaN
## 270                          NaN                      NaN
## 276                          NaN                      NaN
## 290                          NaN                      NaN
## 306                          NaN                      NaN
## 324                          NaN                      NaN
## 331                          NaN                      NaN
## 332                          NaN                      NaN
## 338                          NaN                      NaN
## 342                          NaN                      NaN
## 348                          NaN                      NaN
## 356                          NaN                      NaN
## 366                          NaN                      NaN
## 378                          NaN                      NaN
## 387                          NaN                      NaN
## 401                          NaN                      NaN
## 411                          NaN                      NaN
## 414                          NaN                      NaN
## 421                          NaN                      NaN
## 424                          NaN                      NaN
## 425                          NaN                      NaN
```

```
## 429                         NaN                     NaN
## 431                         NaN                     NaN
## 433                         NaN                     NaN
## 436                         NaN                     NaN
## 439                         NaN                     NaN
## 457                         NaN                     NaN
## 519                         NaN                     NaN
## 542                         NaN                     NaN
## 547                         NaN                     NaN
## 553                         NaN                     NaN
## 559                         NaN                     NaN
## 570                         NaN                     NaN
## 607                         NaN                     NaN
## 611                         NaN                     NaN
## 615                         NaN                     NaN
## 634                         NaN                     NaN
## 645                         NaN                     NaN
## 649                         NaN                     NaN
## 655                         NaN                     NaN
## 656               1682713008576           1682713141741
## 666                         NaN                     NaN
## 685                         NaN                     NaN
## 687                         NaN                     NaN
## 688                         NaN                     NaN
## 691                         NaN                     NaN
##                                                                             other
## 21                                                                           <NA>
## 43                                                                           <NA>
## 78                                                                           <NA>
## 81                                                                           <NA>
## 91                                                                           <NA>
## 94                                                                           <NA>
## 99                                                                           <NA>
## 113                                                                          <NA>
## 136                                                                          <NA>
## 140                                                                          <NA>
## 149                                                                          <NA>
## 177                                                                          <NA>
## 179                                                                          <NA>
## 185                                                                          <NA>
## 186                                                                          <NA>
## 191                                                                          <NA>
## 202                                                                          <NA>
## 211                                                                          <NA>
## 215                                                                         nope.
## 216                                                                          <NA>
## 219                                                                          <NA>
## 236                                                                          <NA>
## 240                                                                          <NA>
## 241                                                                          <NA>
## 254                                                                          <NA>
## 270                                                                          <NA>
## 276                                                                          <NA>
## 290                                                                          <NA>
```

```
## 306                                                                     <NA>
## 324                                                                     <NA>
## 331                                                                     <NA>
## 332 Maybe I could have decided sooner, even. but first round is a lot to go for.
## 338                                                                     <NA>
## 342                                                                     <NA>
## 348                                                                     <NA>
## 356                                                                     <NA>
## 366                                                                     <NA>
## 378                                                                     <NA>
## 387                                                                     <NA>
## 401                                                                     <NA>
## 411                                                                     <NA>
## 414                                                                     <NA>
## 421                                                                     <NA>
## 424                                                                     <NA>
## 425                                                                     <NA>
## 429                                                                     <NA>
## 431                                                                     <NA>
## 433                                                                     <NA>
## 436                                                                     <NA>
## 439                                                                     <NA>
## 457                                                                     <NA>
## 519                                                                     <NA>
## 542                                                                     <NA>
## 547                                                                     <NA>
## 553                                                                     <NA>
## 559                                                                     <NA>
## 570                                                                     <NA>
## 607                                                                     <NA>
## 611                                                                     <NA>
## 615                                                                     <NA>
## 634                                                                     <NA>
## 645                                                                     <NA>
## 649                                                                     <NA>
## 655                                                                     <NA>
## 656                                                                     <NA>
## 666                                                                     <NA>
## 685                                                                     <NA>
## 687                                                                     <NA>
## 688                                                                     <NA>
## 691                                                                     <NA>
##     factual informativeness (comparative).1
## 21                                        2
## 43                                        2
## 78                                        3
## 81                                        3
## 91                                        1
## 94                                        4
## 99                                        1
## 113                                       2
## 136                                       4
## 140                                       4
## 149                                       1
```

```
## 177                                      3
## 179                                    NaN
## 185                                      2
## 186                                      1
## 191                                      1
## 202                                      3
## 211                                      4
## 215                                      3
## 216                                      2
## 219                                      3
## 236                                      3
## 240                                      3
## 241                                      4
## 254                                      3
## 270                                      2
## 276                                      2
## 290                                      2
## 306                                      1
## 324                                      2
## 331                                      2
## 332                                      1
## 338                                      1
## 342                                      3
## 348                                      4
## 356                                      2
## 366                                      1
## 378                                      4
## 387                                      3
## 401                                      4
## 411                                      3
## 414                                      3
## 421                                      1
## 424                                      3
## 425                                      2
## 429                                      3
## 431                                      3
## 433                                      3
## 436                                      3
## 439                                      3
## 457                                    NaN
## 519                                    NaN
## 542                                      3
## 547                                      4
## 553                                      2
## 559                                      3
## 570                                      3
## 607                                      4
## 611                                      2
## 615                                      2
## 634                                      2
## 645                                      3
## 649                                      3
## 655                                      3
## 656                                      1
```

```
## 666                                        2
## 685                                        2
## 687                                        2
## 688                                        2
## 691                                        0
##     factual informativeness (comparative).2 facts versus semantics (single)
## 21                                         2                            NaN
## 43                                         2                            NaN
## 78                                         4                            NaN
## 81                                         3                            NaN
## 91                                         3                            NaN
## 94                                         2                            NaN
## 99                                         3                            NaN
## 113                                        2                            NaN
## 136                                        3                            NaN
## 140                                        3                            NaN
## 149                                        3                            NaN
## 177                                        3                            NaN
## 179                                      NaN                            NaN
## 185                                        2                            NaN
## 186                                        1                            NaN
## 191                                        1                            NaN
## 202                                        4                            NaN
## 211                                        2                            NaN
## 215                                        2                            NaN
## 216                                        2                            NaN
## 219                                        3                            NaN
## 236                                        3                            NaN
## 240                                        3                            NaN
## 241                                        4                            NaN
## 254                                        3                            NaN
## 270                                        3                            NaN
## 276                                        3                            NaN
## 290                                        4                            NaN
## 306                                        0                            NaN
## 324                                        4                            NaN
## 331                                        3                            NaN
## 332                                        4                            NaN
## 338                                        4                            NaN
## 342                                        4                            NaN
## 348                                        4                            NaN
## 356                                        1                            NaN
## 366                                        2                            NaN
## 378                                        4                            NaN
## 387                                        4                            NaN
## 401                                        4                            NaN
## 411                                        3                            NaN
## 414                                        3                            NaN
## 421                                        3                            NaN
## 424                                        3                            NaN
## 425                                        2                            NaN
## 429                                        2                            NaN
## 431                                        3                            NaN
## 433                                        3                            NaN
```

```
## 436                                        3                              NaN
## 439                                        3                              NaN
## 457                                      NaN                              NaN
## 519                                      NaN                              NaN
## 542                                        2                              NaN
## 547                                        4                              NaN
## 553                                        2                              NaN
## 559                                        3                              NaN
## 570                                        3                              NaN
## 607                                        2                              NaN
## 611                                        2                              NaN
## 615                                        3                              NaN
## 634                                        4                              NaN
## 645                                        3                              NaN
## 649                                        1                              NaN
## 655                                        3                              NaN
## 656                                        3                              NaN
## 666                                        3                              NaN
## 685                                        2                              NaN
## 687                                        2                              NaN
## 688                                        1                              NaN
## 691                                        3                              NaN
##     factual accuracy (single) clarity.1 clarity.2 factual accuracy.1
## 21                        NaN         1         1                NaN
## 43                        NaN         2         3                NaN
## 78                        NaN         3         4                NaN
## 81                        NaN         3         3                NaN
## 91                        NaN         1         3                NaN
## 94                        NaN         2         4                NaN
## 99                        NaN         1         3                NaN
## 113                       NaN         2         2                NaN
## 136                       NaN         4         3                NaN
## 140                       NaN         3         3                NaN
## 149                       NaN         1         2                NaN
## 177                       NaN         2         2                NaN
## 179                       NaN       NaN       NaN                NaN
## 185                       NaN         3         4                NaN
## 186                       NaN         3         3                NaN
## 191                       NaN         2         2                NaN
## 202                       NaN         4         4                NaN
## 211                       NaN         4         1                NaN
## 215                       NaN         4         4                NaN
## 216                       NaN         4         4                NaN
## 219                       NaN         3         3                NaN
## 236                       NaN         2         3                NaN
## 240                       NaN         3         3                NaN
## 241                       NaN         4         4                NaN
## 254                       NaN         3         2                NaN
## 270                       NaN         4         4                NaN
## 276                       NaN         3         4                NaN
## 290                       NaN         3         4                NaN
## 306                       NaN         1         0                NaN
## 324                       NaN         0         3                NaN
## 331                       NaN         3         4                NaN
```

```
## 332                       NaN         3         4                 NaN
## 338                       NaN         1         4                 NaN
## 342                       NaN         1         2                 NaN
## 348                       NaN         4         4                 NaN
## 356                       NaN         1         1                 NaN
## 366                       NaN         2         2                 NaN
## 378                       NaN         4         4                 NaN
## 387                       NaN         4         4                 NaN
## 401                       NaN         4         4                 NaN
## 411                       NaN         3         3                 NaN
## 414                       NaN         3         3                 NaN
## 421                       NaN         2         3                 NaN
## 424                       NaN         3         3                 NaN
## 425                       NaN         3         3                 NaN
## 429                       NaN         3         3                 NaN
## 431                       NaN         3         3                 NaN
## 433                       NaN         3         3                 NaN
## 436                       NaN         3         3                 NaN
## 439                       NaN         3         3                 NaN
## 457                       NaN       NaN       NaN                 NaN
## 519                       NaN       NaN       NaN                 NaN
## 542                       NaN         2         3                 NaN
## 547                       NaN         4         4                 NaN
## 553                       NaN         4         4                 NaN
## 559                       NaN         4         4                 NaN
## 570                       NaN         3         3                 NaN
## 607                       NaN         1         4                 NaN
## 611                       NaN         3         2                 NaN
## 615                       NaN         2         3                 NaN
## 634                       NaN         1         4                 NaN
## 645                       NaN         2         2                 NaN
## 649                       NaN         2         1                 NaN
## 655                       NaN         3         3                 NaN
## 656                       NaN         2         3                 NaN
## 666                       NaN         2         2                 NaN
## 685                       NaN         3         2                 NaN
## 687                       NaN         3         2                 NaN
## 688                       NaN         3         1                 NaN
## 691                       NaN         0         3                 NaN
##     factual accuracy.2 judge reasoning
## 21                 NaN               3
## 43                 NaN               3
## 78                 NaN               4
## 81                 NaN               3
## 91                 NaN               4
## 94                 NaN               4
## 99                 NaN               4
## 113                NaN               2
## 136                NaN               4
## 140                NaN               4
## 149                NaN               4
## 177                NaN               3
## 179                NaN             NaN
## 185                NaN               4
```

```
## 186                    NaN                 4
## 191                    NaN                 4
## 202                    NaN                 4
## 211                    NaN                 4
## 215                    NaN                 4
## 216                    NaN                 4
## 219                    NaN                 3
## 236                    NaN                 4
## 240                    NaN                 4
## 241                    NaN                 4
## 254                    NaN                 3
## 270                    NaN                 4
## 276                    NaN                 4
## 290                    NaN                 4
## 306                    NaN                 4
## 324                    NaN                 4
## 331                    NaN                 4
## 332                    NaN                 4
## 338                    NaN                 4
## 342                    NaN                 4
## 348                    NaN                 4
## 356                    NaN                 4
## 366                    NaN                 3
## 378                    NaN                 4
## 387                    NaN                 4
## 401                    NaN                 4
## 411                    NaN                 3
## 414                    NaN                 3
## 421                    NaN               NaN
## 424                    NaN                 3
## 425                    NaN                 3
## 429                    NaN                 3
## 431                    NaN                 3
## 433                    NaN                 3
## 436                    NaN                 4
## 439                    NaN                 3
## 457                    NaN               NaN
## 519                    NaN               NaN
## 542                    NaN                 3
## 547                    NaN                 4
## 553                    NaN                 4
## 559                    NaN                 4
## 570                    NaN                 4
## 607                    NaN                 4
## 611                    NaN                 3
## 615                    NaN                 4
## 634                    NaN                 4
## 645                    NaN                 3
## 649                    NaN                 3
## 655                    NaN                 3
## 656                    NaN                 3
## 666                    NaN                 2
## 685                    NaN                 3
## 687                    NaN                 2
```

```
## 688                 NaN               4
## 691                 NaN               3
##
## 21
## 43
## 78
## 81
## 91
## 94
## 99
## 113
## 136
## 140
## 149
## 177
## 179
## 185
## 186
## 191
## 202
## 211
## 215
## 216
## 219
## 236
## 240
## 241
## 254
## 270
## 276
## 290
## 306
## 324
## 331
## 332
## 338
## 342
## 348
## 356
## 366
## 378                              I think I continued the debate for an extra round just to see if any
## 387
## 401                                                                                Accidentally voted
## 411
## 414
## 421
## 424
## 425
## 429
## 431
## 433
## 436
## 439
## 457
```

```
## 519
## 542
## 547
## 553
## 559
## 570
## 607
## 611
## 615
## 634
## 645
## 649
## 655
## 656
## 666
## 685
## 687
## 688
## 691 I think the factor which convinces me is that the evidence presented seems compelling that the mo
##     protocol evidence use.1 evidence use.2 evidence in story.1
## 21      <NA>            NaN            NaN                  NaN
## 43      <NA>            NaN            NaN                  NaN
## 78      <NA>            NaN            NaN                  NaN
## 81      <NA>            NaN            NaN                  NaN
## 91      <NA>            NaN            NaN                  NaN
## 94      <NA>            NaN            NaN                  NaN
## 99      <NA>            NaN            NaN                  NaN
## 113     <NA>            NaN            NaN                  NaN
## 136     <NA>            NaN            NaN                  NaN
## 140     <NA>            NaN            NaN                  NaN
## 149     <NA>            NaN            NaN                  NaN
## 177     <NA>            NaN            NaN                  NaN
## 179     <NA>            NaN            NaN                  NaN
## 185     <NA>            NaN            NaN                  NaN
## 186     <NA>            NaN            NaN                  NaN
## 191     <NA>            NaN            NaN                  NaN
## 202     <NA>            NaN            NaN                  NaN
## 211     <NA>            NaN            NaN                  NaN
## 215    nope.            NaN            NaN                  NaN
## 216     <NA>            NaN            NaN                  NaN
## 219     <NA>            NaN            NaN                  NaN
## 236     <NA>            NaN            NaN                  NaN
## 240     <NA>            NaN            NaN                  NaN
## 241     <NA>            NaN            NaN                  NaN
## 254     <NA>            NaN            NaN                  NaN
## 270     <NA>            NaN            NaN                  NaN
## 276     <NA>            NaN            NaN                  NaN
## 290     <NA>            NaN            NaN                  NaN
## 306     <NA>            NaN            NaN                  NaN
## 324     <NA>            NaN            NaN                  NaN
## 331     <NA>            NaN            NaN                  NaN
## 332     <NA>            NaN            NaN                  NaN
## 338     <NA>            NaN            NaN                  NaN
## 342     <NA>            NaN            NaN                  NaN
```

```
## 348     <NA>            NaN            NaN                 NaN
## 356     <NA>            NaN            NaN                 NaN
## 366     <NA>            NaN            NaN                 NaN
## 378     <NA>            NaN            NaN                 NaN
## 387     <NA>            NaN            NaN                 NaN
## 401     <NA>            NaN            NaN                 NaN
## 411     <NA>            NaN            NaN                 NaN
## 414     <NA>            NaN            NaN                 NaN
## 421     <NA>            NaN            NaN                 NaN
## 424     <NA>            NaN            NaN                 NaN
## 425     <NA>            NaN            NaN                 NaN
## 429     <NA>            NaN            NaN                 NaN
## 431     <NA>            NaN            NaN                 NaN
## 433     <NA>            NaN            NaN                 NaN
## 436     <NA>            NaN            NaN                 NaN
## 439     <NA>            NaN            NaN                 NaN
## 457     <NA>            NaN            NaN                 NaN
## 519     <NA>            NaN            NaN                 NaN
## 542     <NA>            NaN            NaN                 NaN
## 547     <NA>            NaN            NaN                 NaN
## 553     <NA>            NaN            NaN                 NaN
## 559     <NA>            NaN            NaN                 NaN
## 570     <NA>            NaN            NaN                 NaN
## 607     <NA>            NaN            NaN                 NaN
## 611     <NA>            NaN            NaN                 NaN
## 615     <NA>            NaN            NaN                 NaN
## 634     <NA>            NaN            NaN                 NaN
## 645     <NA>            NaN            NaN                 NaN
## 649     <NA>            NaN            NaN                 NaN
## 655     <NA>            NaN            NaN                 NaN
## 656     <NA>            NaN            NaN                 NaN
## 666     <NA>            NaN            NaN                 NaN
## 685     <NA>            NaN            NaN                 NaN
## 687     <NA>            NaN            NaN                 NaN
## 688     <NA>            NaN            NaN                 NaN
## 691     <NA>            NaN            NaN                 NaN
##     evidence in story.2
## 21                  NaN
## 43                  NaN
## 78                  NaN
## 81                  NaN
## 91                  NaN
## 94                  NaN
## 99                  NaN
## 113                 NaN
## 136                 NaN
## 140                 NaN
## 149                 NaN
## 177                 NaN
## 179                 NaN
## 185                 NaN
## 186                 NaN
## 191                 NaN
## 202                 NaN
```

```
## 211                          NaN
## 215                          NaN
## 216                          NaN
## 219                          NaN
## 236                          NaN
## 240                          NaN
## 241                          NaN
## 254                          NaN
## 270                          NaN
## 276                          NaN
## 290                          NaN
## 306                          NaN
## 324                          NaN
## 331                          NaN
## 332                          NaN
## 338                          NaN
## 342                          NaN
## 348                          NaN
## 356                          NaN
## 366                          NaN
## 378                          NaN
## 387                          NaN
## 401                          NaN
## 411                          NaN
## 414                          NaN
## 421                          NaN
## 424                          NaN
## 425                          NaN
## 429                          NaN
## 431                          NaN
## 433                          NaN
## 436                          NaN
## 439                          NaN
## 457                          NaN
## 519                          NaN
## 542                          NaN
## 547                          NaN
## 553                          NaN
## 559                          NaN
## 570                          NaN
## 607                          NaN
## 611                          NaN
## 615                          NaN
## 634                          NaN
## 645                          NaN
## 649                          NaN
## 655                          NaN
## 656                          NaN
## 666                          NaN
## 685                          NaN
## 687                          NaN
## 688                          NaN
## 691                          NaN
##
```

```
## 21
## 43
## 78
## 81
## 91
## 94
## 99
## 113
## 136
## 140
## 149
## 177
## 179
## 185
## 186
## 191
## 202
## 211
## 215
## 216
## 219
## 236
## 240
## 241
## 254
## 270
## 276
## 290
## 306
## 324
## 331
## 332
## 338
## 342
## 348
## 356
## 366
## 378
## 387
## 401
## 411
## 414
## 421
## 424
## 425
## 429
## 431
## 433
## 436
## 439
## 457
## 519
## 542
## 547
```

```
## 553
## 559
## 570
## 607
## 611
## 615
## 634
## 645
## 649
## 655
## 656
## 666
## 685
## 687 I definitely dropped the ball here and got back to judging the debate after a few weeks. I think
## 688                                              I sensed towards the end that the dishonest debate
## 691
##     judge adaptation (single) evidence in debate.1 evidence in debate.2
## 21                        NaN                    2                    2
## 43                        NaN                    2                    3
## 78                        NaN                    1                    4
## 81                        NaN                    3                    3
## 91                        NaN                    2                    3
## 94                        NaN                    1                    3
## 99                        NaN                    1                    4
## 113                       NaN                    2                    2
## 136                       NaN                    4                    3
## 140                       NaN                    4                    3
## 149                       NaN                    2                    3
## 177                       NaN                    3                    2
## 179                       NaN                  NaN                  NaN
## 185                       NaN                    0                    3
## 186                       NaN                    3                    2
## 191                       NaN                    2                    3
## 202                       NaN                    3                    4
## 211                       NaN                    4                    0
## 215                       NaN                    4                    1
## 216                       NaN                    0                    4
## 219                       NaN                    3                    2
## 236                       NaN                    2                    4
## 240                       NaN                    4                    2
## 241                       NaN                    2                    4
## 254                       NaN                    3                    2
## 270                       NaN                    2                    4
## 276                       NaN                    2                    4
## 290                       NaN                    2                    4
## 306                       NaN                    2                    0
## 324                       NaN                    0                    3
## 331                       NaN                    1                    4
## 332                       NaN                    0                    3
## 338                       NaN                    0                    3
## 342                       NaN                    3                    4
## 348                       NaN                    2                    4
## 356                       NaN                    2                    1
## 366                       NaN                    1                    3
```

```
## 378                          NaN                     4                     4
## 387                          NaN                     2                     4
## 401                          NaN                     4                     4
## 411                          NaN                     3                     3
## 414                          NaN                     3                     2
## 421                          NaN                     1                     3
## 424                          NaN                     2                     3
## 425                          NaN                     3                     3
## 429                          NaN                     3                     3
## 431                          NaN                     4                     3
## 433                          NaN                     3                     2
## 436                          NaN                     3                     3
## 439                          NaN                     2                     3
## 457                          NaN                   NaN                   NaN
## 519                          NaN                   NaN                   NaN
## 542                          NaN                     3                     1
## 547                          NaN                     1                     4
## 553                          NaN                     0                     3
## 559                          NaN                     4                     3
## 570                          NaN                     2                     4
## 607                          NaN                     4                     2
## 611                          NaN                     2                     2
## 615                          NaN                     1                     4
## 634                          NaN                     1                     4
## 645                          NaN                     2                     2
## 649                          NaN                     3                     1
## 655                          NaN                     1                     3
## 656                          NaN                     2                     4
## 666                          NaN                     2                     2
## 685                          NaN                     4                     1
## 687                          NaN                     3                     1
## 688                          NaN                     3                     1
## 691                          NaN                     4                     2
##                                                                  interface
## 21                                                                    <NA>
## 43                                                                    <NA>
## 78                                                                    <NA>
## 81                                                                    <NA>
## 91                                                                    <NA>
## 94                                                                    <NA>
## 99                                                                    <NA>
## 113                                                                   <NA>
## 136                                                                   <NA>
## 140                                                                   <NA>
## 149                                                                   <NA>
## 177                                                                   <NA>
## 179                                                                   <NA>
## 185                    I accidentally entered the probabilities backwards
## 186                                                                   <NA>
## 191                                                                   <NA>
## 202                                                                   <NA>
## 211                                                                   <NA>
## 215                                                 The interface is great!
## 216                                                                   <NA>
```

```
## 219                                                               <NA>
## 236                                                               <NA>
## 240                                                               <NA>
## 241                                                               <NA>
## 254                                                               <NA>
## 270                                                               <NA>
## 276                                                               <NA>
## 290                                                               <NA>
## 306                                                               <NA>
## 324                                                               <NA>
## 331                                                               <NA>
## 332                                                               <NA>
## 338                                                               <NA>
## 342                                                               <NA>
## 348                                                               <NA>
## 356                                                               <NA>
## 366                                                               <NA>
## 378                                                               <NA>
## 387                                                               <NA>
## 401                                                               <NA>
## 411                                                               <NA>
## 414                                                               <NA>
## 421                                                               <NA>
## 424                                                               <NA>
## 425                                                               <NA>
## 429                                                               <NA>
## 431                                                               <NA>
## 433                                                               <NA>
## 436                                                               <NA>
## 439                                                               <NA>
## 457                                                               <NA>
## 519                                                               <NA>
## 542                                                               <NA>
## 547                                                               <NA>
## 553                                                               <NA>
## 559                                                               <NA>
## 570                                                               <NA>
## 607                                                               <NA>
## 611                                                               <NA>
## 615                                                               <NA>
## 634                                                               <NA>
## 645                                                               <NA>
## 649                                                               <NA>
## 655                                                               <NA>
## 656                                                               <NA>
## 666                                                               <NA>
## 685 Quote limits seemed to hamper both debaters? Unclear if they agree
## 687                                                               <NA>
## 688                                                               <NA>
## 691                                                               <NA>
##     evidence in debate (single) facts versus semantics.1
## 21                        NaN                        3
## 43                        NaN                        3
## 78                        NaN                        2
```

```
## 81                              NaN                         1
## 91                              NaN                         3
## 94                              NaN                         2
## 99                              NaN                         2
## 113                             NaN                         2
## 136                             NaN                         4
## 140                             NaN                         1
## 149                             NaN                         3
## 177                             NaN                         1
## 179                             NaN                       NaN
## 185                             NaN                         2
## 186                             NaN                         2
## 191                             NaN                         3
## 202                             NaN                         0
## 211                             NaN                         0
## 215                             NaN                         0
## 216                             NaN                         0
## 219                             NaN                         2
## 236                             NaN                         1
## 240                             NaN                         1
## 241                             NaN                         0
## 254                             NaN                         1
## 270                             NaN                         2
## 276                             NaN                         2
## 290                             NaN                         3
## 306                             NaN                         4
## 324                             NaN                         0
## 331                             NaN                         0
## 332                             NaN                         0
## 338                             NaN                         0
## 342                             NaN                         0
## 348                             NaN                         3
## 356                             NaN                         1
## 366                             NaN                         2
## 378                             NaN                         3
## 387                             NaN                         4
## 401                             NaN                         1
## 411                             NaN                         3
## 414                             NaN                         1
## 421                             NaN                         1
## 424                             NaN                         1
## 425                             NaN                         2
## 429                             NaN                         1
## 431                             NaN                         1
## 433                             NaN                         1
## 436                             NaN                         2
## 439                             NaN                         2
## 457                             NaN                       NaN
## 519                             NaN                       NaN
## 542                             NaN                         2
## 547                             NaN                         3
## 553                             NaN                         3
## 559                             NaN                         2
## 570                             NaN                         2
```

```
## 607                          NaN                         2
## 611                          NaN                         2
## 615                          NaN                         2
## 634                          NaN                         0
## 645                          NaN                         1
## 649                          NaN                         1
## 655                          NaN                         3
## 656                          NaN                         2
## 666                          NaN                         3
## 685                          NaN                         1
## 687                          NaN                         2
## 688                          NaN                         2
## 691                          NaN                         1
##     facts versus semantics.2 clash.1 clash.2 identity guesses.Judge
## 21                         3       1       2                   <NA>
## 43                         2       2       2                   <NA>
## 78                         1       1       4                   <NA>
## 81                         1       3       3                   <NA>
## 91                         1       0       3                   <NA>
## 94                         1       1       4                   <NA>
## 99                         3       1       3                   <NA>
## 113                        2       2       2                   <NA>
## 136                        3       4       3                   <NA>
## 140                        3       3       2                   <NA>
## 149                        1       2       2                   <NA>
## 177                        3       2       2                   <NA>
## 179                      NaN     NaN     NaN                   <NA>
## 185                        1       2       3                   <NA>
## 186                        1       4       2                   <NA>
## 191                        3       3       3                   <NA>
## 202                        0       3       4                   <NA>
## 211                        4       4       1                   <NA>
## 215                        1       3       2                   <NA>
## 216                        0       0       4                   <NA>
## 219                        2       3       2                   <NA>
## 236                        1       3       3                   <NA>
## 240                        1       4       3                   <NA>
## 241                        0       3       4                   <NA>
## 254                        3       3       2                   <NA>
## 270                        0       2       2                   <NA>
## 276                        0       2       4                   <NA>
## 290                        0       1       2                   <NA>
## 306                        3       1       0                   <NA>
## 324                        0       1       4                   <NA>
## 331                        0       1       4                   <NA>
## 332                        0       0       4                   <NA>
## 338                        0       0       4                   <NA>
## 342                        0       2       4                   <NA>
## 348                        1       4       4                   <NA>
## 356                        2       1       1                   <NA>
## 366                        1       2       3                   <NA>
## 378                        0       4       4                   <NA>
## 387                        4       4       4                   <NA>
## 401                        0       4       4                   <NA>
```

```
## 411                        2        3        3                 <NA>
## 414                        2        3        3                 <NA>
## 421                        1        2        3                 <NA>
## 424                        1        3        3                 <NA>
## 425                        2        3        3                 <NA>
## 429                        1        3        2                 <NA>
## 431                        2        3        2                 <NA>
## 433                        1        3        3                 <NA>
## 436                        2        3        3                 <NA>
## 439                        1        2        2                 <NA>
## 457                      NaN      NaN      NaN                 <NA>
## 519                      NaN      NaN      NaN                 <NA>
## 542                        1        3        3                 <NA>
## 547                        3        4        2                 <NA>
## 553                        0        0        0                 <NA>
## 559                        1        4        2                 <NA>
## 570                        0        4        4                 <NA>
## 607                        1        2        2                 <NA>
## 611                        2        3        3                 <NA>
## 615                        2        3        2                 <NA>
## 634                        0        0        4                 <NA>
## 645                        3        3        3                 <NA>
## 649                        3        2        0                 <NA>
## 655                        2        2        3                 <NA>
## 656                        1        2        4                 <NA>
## 666                        3        1        3                 <NA>
## 685                        3        4        2                 <NA>
## 687                        3        3        2                 <NA>
## 688                        3        4        2                 <NA>
## 691                        3        1        3                 <NA>
##     identity guesses.Debater A identity guesses.Debater B judge adaptation.1
## 21                        <NA>                       <NA>                  1
## 43                        <NA>                       <NA>                  2
## 78                        <NA>                       <NA>                  2
## 81                        <NA>                       <NA>                  3
## 91                        <NA>                       <NA>                  2
## 94                        <NA>                       <NA>                  1
## 99                        <NA>                       <NA>                  0
## 113                       <NA>                       <NA>                  2
## 136                       <NA>                       <NA>                  4
## 140                       <NA>                       <NA>                  3
## 149          Emmanuel Makinde                        <NA>                  1
## 177                       <NA>                       <NA>                  1
## 179                       <NA>                       <NA>                NaN
## 185                       <NA>                       <NA>                  2
## 186                       <NA>                       <NA>                  2
## 191                       <NA>                       <NA>                  4
## 202                       <NA>                       <NA>                  2
## 211                       <NA>                       <NA>                  4
## 215                       <NA>                       <NA>                  4
## 216                       <NA>                       <NA>                  2
## 219                       <NA>                       <NA>                  3
## 236                       <NA>                       <NA>                  3
## 240                       <NA>                       <NA>                  4
```

```
## 241                       <NA>                       <NA>                  2
## 254                       <NA>                       <NA>                  3
## 270                       <NA>                       <NA>                  3
## 276                       <NA>                       <NA>                  2
## 290                       <NA>                       <NA>                  2
## 306                       <NA>                       <NA>                  1
## 324               Reeya Kansra                  Sean Wang                  2
## 331               Reeya Kansra                  Sean Wang                  0
## 332                       <NA>                       <NA>                  4
## 338                       <NA>                       <NA>                  0
## 342                       <NA>                       <NA>                  2
## 348                       <NA>                       <NA>                  3
## 356                       <NA>                       <NA>                  2
## 366                       <NA>                       <NA>                  3
## 378                 Jessica Li            Adelle Fernando                  4
## 387              Julien Dirani                Ethan Rosen                  4
## 401           Emmanuel Makinde            Adelle Fernando                  4
## 411                       <NA>                       <NA>                  3
## 414                       <NA>                       <NA>                  2
## 421                       <NA>                       <NA>                  3
## 424              Shlomo Kofman                    Sam Jin                  3
## 425                 Jessica Li                  Anuj Jain                  3
## 429                 Jessica Li              Shreeram Modi                  3
## 431                       <NA>                       <NA>                  3
## 433                       <NA>                       <NA>                  3
## 436                       <NA>                       <NA>                  4
## 439                  Sean Wang               Reeya Kansra                  3
## 457                       <NA>                       <NA>                NaN
## 519                       <NA>                       <NA>                NaN
## 542                       <NA>                       <NA>                  2
## 547                       <NA>                       <NA>                  4
## 553                       <NA>                       <NA>                  2
## 559                       <NA>                       <NA>                  3
## 570                       <NA>                       <NA>                  3
## 607                       <NA>                       <NA>                  2
## 611                       <NA>                       <NA>                  3
## 615                       <NA>                       <NA>                  2
## 634                       <NA>                       <NA>                  0
## 645                       <NA>                       <NA>                  3
## 649                       <NA>                       <NA>                  2
## 655                       <NA>                       <NA>                  2
## 656                       <NA>                       <NA>                  1
## 666                       <NA>                       <NA>                  1
## 685                       <NA>                       <NA>                  4
## 687                       <NA>                       <NA>                  3
## 688                       <NA>                       <NA>                  4
## 691                       <NA>                       <NA>                  1
##     judge adaptation.2 subjective correctness evidence use (single)
## 21                   1                    NaN                   NaN
## 43                   3                    NaN                   NaN
## 78                   4                    NaN                   NaN
## 81                   3                    NaN                   NaN
## 91                   3                    NaN                   NaN
## 94                   4                    NaN                   NaN
```

```
## 99                    4                    NaN                    NaN
## 113                   2                    NaN                    NaN
## 136                   3                    NaN                    NaN
## 140                   2                    NaN                    NaN
## 149                   2                    NaN                    NaN
## 177                   1                    NaN                    NaN
## 179                 NaN                    NaN                    NaN
## 185                   2                    NaN                    NaN
## 186                   2                    NaN                    NaN
## 191                   4                    NaN                    NaN
## 202                   4                    NaN                    NaN
## 211                   1                    NaN                    NaN
## 215                   4                    NaN                    NaN
## 216                   4                    NaN                    NaN
## 219                   2                    NaN                    NaN
## 236                   4                    NaN                    NaN
## 240                   2                    NaN                    NaN
## 241                   4                    NaN                    NaN
## 254                   1                    NaN                    NaN
## 270                   4                    NaN                    NaN
## 276                   4                    NaN                    NaN
## 290                   2                    NaN                    NaN
## 306                   0                    NaN                    NaN
## 324                   4                    NaN                    NaN
## 331                   4                    NaN                    NaN
## 332                   4                    NaN                    NaN
## 338                   4                    NaN                    NaN
## 342                   3                    NaN                    NaN
## 348                   3                    NaN                    NaN
## 356                   2                    NaN                    NaN
## 366                   3                    NaN                    NaN
## 378                   4                    NaN                    NaN
## 387                   4                    NaN                    NaN
## 401                   4                    NaN                    NaN
## 411                   3                    NaN                    NaN
## 414                   3                    NaN                    NaN
## 421                   4                    NaN                    NaN
## 424                   3                    NaN                    NaN
## 425                   3                    NaN                    NaN
## 429                   2                    NaN                    NaN
## 431                   3                    NaN                    NaN
## 433                   2                    NaN                    NaN
## 436                   3                    NaN                    NaN
## 439                   3                    NaN                    NaN
## 457                 NaN                    NaN                    NaN
## 519                 NaN                    NaN                    NaN
## 542                   3                    NaN                    NaN
## 547                   4                    NaN                    NaN
## 553                   2                    NaN                    NaN
## 559                   3                    NaN                    NaN
## 570                   3                    NaN                    NaN
## 607                   2                    NaN                    NaN
## 611                   3                    NaN                    NaN
## 615                   3                    NaN                    NaN
```

```
## 634                 4                   NaN                  NaN
## 645                 3                   NaN                  NaN
## 649                 1                   NaN                  NaN
## 655                 2                   NaN                  NaN
## 656                 4                   NaN                  NaN
## 666                 3                   NaN                  NaN
## 685                 1                   NaN                  NaN
## 687                 0                   NaN                  NaN
## 688                 2                   NaN                  NaN
## 691                 3                   NaN                  NaN
##     factual informativeness (total)
## 21                                1
## 43                                2
## 78                                3
## 81                                3
## 91                                3
## 94                                3
## 99                                3
## 113                               2
## 136                               4
## 140                               3
## 149                               3
## 177                               1
## 179                             NaN
## 185                               1
## 186                               1
## 191                               1
## 202                               4
## 211                               3
## 215                               2
## 216                               0
## 219                               3
## 236                               4
## 240                               4
## 241                               4
## 254                               3
## 270                               3
## 276                               4
## 290                               3
## 306                               0
## 324                               4
## 331                               3
## 332                               4
## 338                               3
## 342                               3
## 348                               3
## 356                               2
## 366                               2
## 378                               4
## 387                               4
## 401                               4
## 411                               3
## 414                               3
## 421                               3
```

```
## 424                                   3
## 425                                   3
## 429                                   3
## 431                                   3
## 433                                   3
## 436                                   4
## 439                                   3
## 457                                 NaN
## 519                                 NaN
## 542                                   3
## 547                                   4
## 553                                   0
## 559                                   3
## 570                                   4
## 607                                   4
## 611                                   3
## 615                                   3
## 634                                   4
## 645                                   3
## 649                                   2
## 655                                   3
## 656                                   3
## 666                                   3
## 685                                   1
## 687                                   2
## 688                                   3
## 691                                   3
##
## 21
## 43
## 78
## 81
## 91
## 94
## 99
## 113
## 136
## 140
## 149
## 177
## 179
## 185                        I said this to debater A: Are there any other resources mentioned, or context
## 186
## 191
## 202
## 211
## 215
## 216
## 219
## 236
## 240
## 241
## 254
## 270
```

```
## 276
## 290
## 306
## 324
## 331
## 332
## 338
## 342
## 348
## 356
## 366
## 378
## 387
## 401
## 411
## 414
## 421
## 424
## 425
## 429
## 431
## 433
## 436
## 439
## 457
## 519
## 542
## 547
## 553
## 559
## 570
## 607
## 611
## 615
## 634
## 645
## 649
## 655
## 656
## 666 Yes. I indicated particular pieces of evidence that both were missing and that would help me grea
## 685
## 687
## 688
## 691
##     clarity (single)          Debater A          Debater B     Honest debater
## 21               NaN        Ethan Rosen          Sean Wang        Ethan Rosen
## 43               NaN         Jessica Li        Ethan Rosen        Ethan Rosen
## 78               NaN       Reeya Kansra     Julian Michael     Julian Michael
## 81               NaN      Shreeram Modi          Sean Wang      Shreeram Modi
## 91               NaN      Shlomo Kofman          Sean Wang          Sean Wang
## 94               NaN          Sean Wang          Anuj Jain          Anuj Jain
## 99               NaN    Adelle Fernando      Shreeram Modi      Shreeram Modi
## 113              NaN  Noor Mirza-Rashid          Sean Wang  Noor Mirza-Rashid
## 136              NaN      Shreeram Modi    Adelle Fernando      Shreeram Modi
```

```
## 140               NaN       Reeya Kansra        Jessica Li       Reeya Kansra
## 149               NaN    Salsabila Mahdi        Jessica Li         Jessica Li
## 177               NaN        Ethan Rosen      Reeya Kansra        Ethan Rosen
## 179               NaN       Reeya Kansra     Jackson Petty      Jackson Petty
## 185               NaN      Shreeram Modi       Ethan Rosen        Ethan Rosen
## 186               NaN      Shreeram Modi   Adelle Fernando      Shreeram Modi
## 191               NaN          Sean Wang   Salsabila Mahdi    Salsabila Mahdi
## 202               NaN    Adelle Fernando         Sean Wang          Sean Wang
## 211               NaN          Sean Wang     Shreeram Modi          Sean Wang
## 215               NaN         David Rein         Sean Wang         David Rein
## 216               NaN  Noor Mirza-Rashid     Shreeram Modi      Shreeram Modi
## 219               NaN          Anuj Jain           Sam Jin          Anuj Jain
## 236               NaN            Sam Jin     Shlomo Kofman      Shlomo Kofman
## 240               NaN    Adelle Fernando      Reeya Kansra    Adelle Fernando
## 241               NaN            Sam Jin         Anuj Jain          Anuj Jain
## 254               NaN          Anuj Jain      Reeya Kansra          Anuj Jain
## 270               NaN       Reeya Kansra         Anuj Jain          Anuj Jain
## 276               NaN    Adelle Fernando       Ethan Rosen        Ethan Rosen
## 290               NaN    Adelle Fernando           Sam Jin            Sam Jin
## 306               NaN          Anuj Jain         Sean Wang          Anuj Jain
## 324               NaN       Reeya Kansra         Sean Wang          Sean Wang
## 331               NaN      Shreeram Modi         Sean Wang          Sean Wang
## 332               NaN    Adelle Fernando       Ethan Rosen        Ethan Rosen
## 338               NaN      Shreeram Modi         Anuj Jain          Anuj Jain
## 342               NaN         Jessica Li         Anuj Jain          Anuj Jain
## 348               NaN          Sean Wang        Jessica Li         Jessica Li
## 356               NaN       Reeya Kansra     Shreeram Modi       Reeya Kansra
## 366               NaN      Shreeram Modi   Salsabila Mahdi    Salsabila Mahdi
## 378               NaN    Salsabila Mahdi       Ethan Rosen        Ethan Rosen
## 387               NaN            Sam Jin     Shlomo Kofman      Shlomo Kofman
## 401               NaN          Anuj Jain Noor Mirza-Rashid          Anuj Jain
## 411               NaN          Sean Wang         Anuj Jain          Anuj Jain
## 414               NaN          Sean Wang   Adelle Fernando    Adelle Fernando
## 421               NaN      Shreeram Modi    Julian Michael     Julian Michael
## 424               NaN      Shlomo Kofman           Sam Jin            Sam Jin
## 425               NaN         Jessica Li     Shreeram Modi         Jessica Li
## 429               NaN            Sam Jin   Adelle Fernando            Sam Jin
## 431               NaN      Shreeram Modi   Adelle Fernando      Shreeram Modi
## 433               NaN  Aliyaah Toussaint   Adelle Fernando  Aliyaah Toussaint
## 436               NaN Vishakh Padmakumar     Shreeram Modi Vishakh Padmakumar
## 439               NaN            Sam Jin   Adelle Fernando    Adelle Fernando
## 457               NaN    Adelle Fernando        Jessica Li    Adelle Fernando
## 519               NaN          Anuj Jain     Shlomo Kofman          Anuj Jain
## 542               NaN      Shreeram Modi   Salsabila Mahdi    Salsabila Mahdi
## 547               NaN    Salsabila Mahdi Vishakh Padmakumar Vishakh Padmakumar
## 553               NaN    Adelle Fernando     Jackson Petty      Jackson Petty
## 559               NaN         Jessica Li     Shlomo Kofman         Jessica Li
## 570               NaN      Shreeram Modi Aliyaah Toussaint  Aliyaah Toussaint
## 607               NaN      Shreeram Modi   Adelle Fernando    Adelle Fernando
## 611               NaN         Jessica Li         Sean Wang         Jessica Li
## 615               NaN    Adelle Fernando      Reeya Kansra       Reeya Kansra
## 634               NaN            Sam Jin   Adelle Fernando    Adelle Fernando
## 645               NaN          Sean Wang   Adelle Fernando          Sean Wang
## 649               NaN         David Rein        Jessica Li         David Rein
```

```
## 655             NaN    Salsabila Mahdi Vishakh Padmakumar Vishakh Padmakumar
## 656             NaN          Sean Wang        Ethan Rosen        Ethan Rosen
## 666             NaN          Sean Wang     Julian Michael     Julian Michael
## 685             NaN     Julian Michael         Jessica Li     Julian Michael
## 687             NaN         Jessica Li    Salsabila Mahdi    Salsabila Mahdi
## 688             NaN     Julian Michael    Salsabila Mahdi     Julian Michael
## 691             NaN          Anuj Jain      Shreeram Modi          Anuj Jain
##     Dishonest debater Is single debater Has honest debater Final_Setting
## 21          Sean Wang             FALSE               TRUE  Human Debate
## 43         Jessica Li             FALSE               TRUE  Human Debate
## 78       Reeya Kansra             FALSE               TRUE  Human Debate
## 81          Sean Wang             FALSE               TRUE  Human Debate
## 91       Shlomo Kofman            FALSE               TRUE  Human Debate
## 94          Sean Wang             FALSE               TRUE  Human Debate
## 99    Adelle Fernando             FALSE               TRUE  Human Debate
## 113         Sean Wang             FALSE               TRUE  Human Debate
## 136   Adelle Fernando             FALSE               TRUE  Human Debate
## 140        Jessica Li             FALSE               TRUE  Human Debate
## 149   Salsabila Mahdi             FALSE               TRUE  Human Debate
## 177      Reeya Kansra             FALSE               TRUE  Human Debate
## 179      Reeya Kansra             FALSE               TRUE  Human Debate
## 185     Shreeram Modi             FALSE               TRUE  Human Debate
## 186   Adelle Fernando             FALSE               TRUE  Human Debate
## 191         Sean Wang             FALSE               TRUE  Human Debate
## 202   Adelle Fernando             FALSE               TRUE  Human Debate
## 211     Shreeram Modi             FALSE               TRUE  Human Debate
## 215         Sean Wang             FALSE               TRUE  Human Debate
## 216 Noor Mirza-Rashid             FALSE               TRUE  Human Debate
## 219           Sam Jin             FALSE               TRUE  Human Debate
## 236           Sam Jin             FALSE               TRUE  Human Debate
## 240      Reeya Kansra             FALSE               TRUE  Human Debate
## 241           Sam Jin             FALSE               TRUE  Human Debate
## 254      Reeya Kansra             FALSE               TRUE  Human Debate
## 270      Reeya Kansra             FALSE               TRUE  Human Debate
## 276   Adelle Fernando             FALSE               TRUE  Human Debate
## 290   Adelle Fernando             FALSE               TRUE  Human Debate
## 306         Sean Wang             FALSE               TRUE  Human Debate
## 324      Reeya Kansra             FALSE               TRUE  Human Debate
## 331     Shreeram Modi             FALSE               TRUE  Human Debate
## 332   Adelle Fernando             FALSE               TRUE  Human Debate
## 338     Shreeram Modi             FALSE               TRUE  Human Debate
## 342        Jessica Li             FALSE               TRUE  Human Debate
## 348         Sean Wang             FALSE               TRUE  Human Debate
## 356     Shreeram Modi             FALSE               TRUE  Human Debate
## 366     Shreeram Modi             FALSE               TRUE  Human Debate
## 378   Salsabila Mahdi             FALSE               TRUE  Human Debate
## 387           Sam Jin             FALSE               TRUE  Human Debate
## 401 Noor Mirza-Rashid             FALSE               TRUE  Human Debate
## 411         Sean Wang             FALSE               TRUE  Human Debate
## 414         Sean Wang             FALSE               TRUE  Human Debate
## 421     Shreeram Modi             FALSE               TRUE  Human Debate
## 424     Shlomo Kofman             FALSE               TRUE  Human Debate
## 425     Shreeram Modi             FALSE               TRUE  Human Debate
## 429   Adelle Fernando             FALSE               TRUE  Human Debate
```

```
## 431   Adelle Fernando              FALSE               TRUE  Human Debate
## 433   Adelle Fernando              FALSE               TRUE  Human Debate
## 436     Shreeram Modi              FALSE               TRUE  Human Debate
## 439           Sam Jin              FALSE               TRUE  Human Debate
## 457        Jessica Li              FALSE               TRUE  Human Debate
## 519     Shlomo Kofman              FALSE               TRUE  Human Debate
## 542     Shreeram Modi              FALSE               TRUE  Human Debate
## 547   Salsabila Mahdi              FALSE               TRUE  Human Debate
## 553   Adelle Fernando              FALSE               TRUE  Human Debate
## 559     Shlomo Kofman              FALSE               TRUE  Human Debate
## 570     Shreeram Modi              FALSE               TRUE  Human Debate
## 607     Shreeram Modi              FALSE               TRUE  Human Debate
## 611         Sean Wang              FALSE               TRUE  Human Debate
## 615   Adelle Fernando              FALSE               TRUE  Human Debate
## 634           Sam Jin              FALSE               TRUE  Human Debate
## 645   Adelle Fernando              FALSE               TRUE  Human Debate
## 649        Jessica Li              FALSE               TRUE  Human Debate
## 655   Salsabila Mahdi              FALSE               TRUE  Human Debate
## 656         Sean Wang              FALSE               TRUE  Human Debate
## 666         Sean Wang              FALSE               TRUE  Human Debate
## 685        Jessica Li              FALSE               TRUE  Human Debate
## 687        Jessica Li              FALSE               TRUE  Human Debate
## 688   Salsabila Mahdi              FALSE               TRUE  Human Debate
## 691     Shreeram Modi              FALSE               TRUE  Human Debate
##          Setting
## 21  Human Debate
## 43  Human Debate
## 78  Human Debate
## 81  Human Debate
## 91  Human Debate
## 94  Human Debate
## 99  Human Debate
## 113 Human Debate
## 136 Human Debate
## 140 Human Debate
## 149 Human Debate
## 177 Human Debate
## 179 Human Debate
## 185 Human Debate
## 186 Human Debate
## 191 Human Debate
## 202 Human Debate
## 211 Human Debate
## 215 Human Debate
## 216 Human Debate
## 219 Human Debate
## 236 Human Debate
## 240 Human Debate
## 241 Human Debate
## 254 Human Debate
## 270 Human Debate
## 276 Human Debate
## 290 Human Debate
## 306 Human Debate
```

```
## 324 Human Debate
## 331 Human Debate
## 332 Human Debate
## 338 Human Debate
## 342 Human Debate
## 348 Human Debate
## 356 Human Debate
## 366 Human Debate
## 378 Human Debate
## 387 Human Debate
## 401 Human Debate
## 411 Human Debate
## 414 Human Debate
## 421 Human Debate
## 424 Human Debate
## 425 Human Debate
## 429 Human Debate
## 431 Human Debate
## 433 Human Debate
## 436 Human Debate
## 439 Human Debate
## 457 Human Debate
## 519 Human Debate
## 542 Human Debate
## 547 Human Debate
## 553 Human Debate
## 559 Human Debate
## 570 Human Debate
## 607 Human Debate
## 611 Human Debate
## 615 Human Debate
## 634 Human Debate
## 645 Human Debate
## 649 Human Debate
## 655 Human Debate
## 656 Human Debate
## 666 Human Debate
## 685 Human Debate
## 687 Human Debate
## 688 Human Debate
## 691 Human Debate
##
## 21                                                                    Which is
## 43                                                                      Which
## 78                                                                How did Earth
## 81                                                             Why does Koroby
## 91                                                       Did the questions Tremain
## 94                                                        Why did the physicist an
## 99                                                                What was the bl
## 113                                                                   What is li
## 136                                                                     Why was
## 140
## 149                                                 Why was the main character daydream
## 177                                                 Generally, which of the following b
```

```
## 179 Which of these sets of de
## 185 What was the
## 186
## 191
## 202 Why do Bob and Quezy ha
## 211
## 215 Why was Dr. Crander so
## 216 What is not a type techno
## 219 What best describes how the overall tone changed f
## 236 What would best describe Asa's
## 240 Why did the Earth
## 241 Who are the four to blame
## 254 What did Zen think of the plan the r
## 270 Why is Grannie Annie so concerned abou
## 276 How many compan
## 290 Who are the four to blame
## 306 What was the population o
## 324 Which i
## 331 Why does Koroby
## 332 How did the planet of Niobe compare to
## 338 Which best describes the relat
## 342 What is the relationship between
## 348 What is the
## 356 Why is
## 366 What does the narrat
## 378
## 387 What happens to a changelin
## 401 What makes the protagonists become less concerned a
## 411 Why was the approach that Charlie took to eng
## 414 Why do Bob and Quezy ha
## 421 Why is Jor
## 424 What would best describe Asa's
## 425
## 429 What is Androka's motivation
## 431 Which of the following is not a reason why Koroby is impressed by the s
## 433 Johnathan doesn't tell the Interstellar Cosmography Society about the twenty-seven women who are
## 436 How many people
## 439 What was the bl
## 457
## 519 What was the relationship like between
## 542 Why did Korvin have to word his
## 547 How would you describe
## 553
## 559 Why did Pashk
## 570 What does the narrat
## 607 Why wa
## 611 What would be the main reason Mr. Ranson wants to find t
## 615 Why did the Earth
## 634 How did H
## 645 What is l
## 649 What is likely to happen to the crew
## 655 Wh
## 656 What is the true explanation for Charles
## 666 If Dan and Erica had been seen together before the accident, what
```

```
## 685                                                     Of the following situations, what was th
## 687                                                         Why was the main character daydreami
## 688                                                                        What is the style of
## 691                                                                    What is not a type techno
##     Article ID Story length Speed annotator accuracy bins
## 21       61499        24253                             0
## 43       61053        23329                             0
## 78       60412        22224                             0
## 81       62314        21057                           0.2
## 91       52844        22871                           0.2
## 94       51126        25560                           0.2
## 99       52855        24058                           0.2
## 113      63527        24795                             0
## 136      63633        21817                           0.2
## 140      43046        25243                           0.4
## 149      51688        24411                           0.2
## 177      61499        24253                           0.2
## 179      55933        20675                           0.4
## 185      63862        23765                           0.2
## 186      62314        21057                           0.2
## 191      50893        21233                           0.2
## 202      63527        24795                           0.2
## 211      62314        21057                           0.2
## 215      51295        24055                           0.4
## 216      62569        24855                           0.4
## 219      53269        26147                           0.2
## 236      61467        21862                           0.4
## 240      60412        22224                           0.2
## 241      61481        23091                           0.2
## 254      51126        25560                             0
## 270      63109        21042                           0.2
## 276      50818        24698                           0.2
## 290      61481        23091                           0.2
## 306      51126        25560                           0.2
## 324      61499        24253                             0
## 331      62314        21057                           0.2
## 332      51395        25010                           0.2
## 338      62569        24855                           0.2
## 342      51351        26909                           0.2
## 348      61430        24002                             0
## 356      63109        21042                           0.2
## 366      51201        24730                             0
## 378      50818        24698                           0.4
## 387      61467        21862                           0.4
## 401      62569        24855                             0
## 411      51320        23858                           0.2
## 414      63527        24795                           0.2
## 421      61430        24002                           0.4
## 424      61467        21862                           0.4
## 425      43046        25243                           0.4
## 429      61481        23091                             0
## 431      62314        21057                           0.2
## 433      63401        20713                           0.2
## 436      51483        22857                           0.2
```

```
## 439      52855        24058                               0.2
## 457      63523        22622                               0.2
## 519      51150        23018                               0.2
## 542      30029        20674                               0.4
## 547      55933        20675                                 0
## 553      51395        25010                               0.2
## 559      51256        26921                               0.4
## 570      51201        24730                                 0
## 607      63633        21817                               0.2
## 611      62085        20786                               0.2
## 615      60412        22224                               0.2
## 634      63855        24457                                 0
## 645      63527        24795                                 0
## 649      61007        15499                               0.2
## 655      55933        20675                               0.2
## 656      63631        20259                               0.2
## 666      51295        24055                               0.4
## 685      63862        23765                               0.4
## 687      51688        24411                               0.2
## 688      61285        24640                               0.4
## 691      62569        24855                               0.4
##     Untimed annotator context bins Speed annotator accuracy
## 21                               4                0.0000000
## 43                               4                0.0000000
## 78                               2                0.0000000
## 81                               3                0.2000000
## 91                               4                0.2000000
## 94                               2                0.2000000
## 99                               3                0.2000000
## 113                              3                0.0000000
## 136                              4                0.2000000
## 140                              2                0.4000000
## 149                              2                0.2000000
## 177                              3                0.2000000
## 179                              3                0.4000000
## 185                              2                0.2000000
## 186                              3                0.2000000
## 191                              3                0.2000000
## 202                              2                0.2000000
## 211                              3                0.2000000
## 215                              3                0.4000000
## 216                              3                0.4000000
## 219                              4                0.2000000
## 236                              2                0.4000000
## 240                              3                0.2000000
## 241                              3                0.2000000
## 254                              2                0.0000000
## 270                              2                0.2000000
## 276                              3                0.2000000
## 290                              3                0.2000000
## 306                              2                0.2000000
## 324                              4                0.0000000
## 331                              3                0.2000000
## 332                              3                0.1666667
```

```
## 338                              3             0.2000000
## 342                              3             0.1666667
## 348                              2             0.0000000
## 356                              3             0.2000000
## 366                              3             0.0000000
## 378                              4             0.4000000
## 387                              2             0.4000000
## 401                              2             0.0000000
## 411                              2             0.1666667
## 414                              2             0.2000000
## 421                              2             0.4000000
## 424                              2             0.4000000
## 425                              3             0.4000000
## 429                              3             0.0000000
## 431                              2             0.2000000
## 433                              2             0.2000000
## 436                              2             0.2000000
## 439                              3             0.2000000
## 457                              3             0.2000000
## 519                              3             0.2000000
## 542                              2             0.4000000
## 547                              4             0.0000000
## 553                              2             0.2000000
## 559                              3             0.4000000
## 570                              3             0.0000000
## 607                              4             0.2000000
## 611                              2             0.2000000
## 615                              3             0.2000000
## 634                              2             0.0000000
## 645                              3             0.0000000
## 649                              2             0.2000000
## 655                              2             0.2000000
## 656                              3             0.2000000
## 666                              4             0.4000000
## 685                              3             0.4000000
## 687                              2             0.2000000
## 688                              2             0.4000000
## 691                              3             0.4000000
##     Untimed annotator context Is offline            End time
## 21                   3.666667      FALSE 2023-04-10 16:16:41
## 43                   3.666667      FALSE 2023-05-21 14:03:16
## 78                   2.000000      FALSE 2023-05-19 15:40:18
## 81                   3.000000      FALSE 2023-06-22 17:38:01
## 91                   4.000000      FALSE 2023-07-27 16:36:48
## 94                   1.800000      FALSE 2023-06-29 18:36:11
## 99                   2.600000      FALSE 2023-07-13 17:57:20
## 113                  3.000000      FALSE 2023-04-21 16:43:34
## 136                  4.000000      FALSE 2023-07-24 15:45:08
## 140                  1.600000      FALSE 2023-04-17 16:40:55
## 149                  2.333333      FALSE 2023-04-10 17:33:21
## 177                  3.333333      FALSE 2023-04-18 15:05:57
## 179                  3.333333      FALSE 2023-07-20 15:41:51
## 185                  2.000000      FALSE 2023-02-27 17:02:34
## 186                  2.600000      FALSE 2023-05-12 16:09:16
```

```
## 191                      3.333333      FALSE 2023-05-09 16:15:12
## 202                      1.666667      FALSE 2023-04-14 18:04:29
## 211                      2.600000      FALSE 2023-05-12 16:15:12
## 215                      3.000000      FALSE 2023-02-13 16:41:56
## 216                      3.000000      FALSE 2023-04-14 16:31:19
## 219                      3.666667      FALSE 2023-07-28 15:39:59
## 236                      2.333333      FALSE 2023-06-26 17:15:36
## 240                      2.600000      FALSE 2023-06-16 16:50:59
## 241                      3.333333      FALSE 2023-07-17 16:33:07
## 254                      2.200000      FALSE 2023-07-17 15:04:00
## 270                      1.666667      FALSE 2023-04-14 17:10:57
## 276                      3.400000      FALSE 2023-05-15 16:10:35
## 290                      3.333333      FALSE 2023-07-06 15:47:04
## 306                      2.200000      FALSE 2023-06-29 17:10:29
## 324                      3.666667      FALSE 2023-05-01 17:55:02
## 331                      3.000000      FALSE 2023-05-05 11:55:03
## 332                      2.750000      FALSE 2023-04-15 06:30:53
## 338                      3.000000      FALSE 2023-06-22 18:58:39
## 342                      2.800000      FALSE 2023-06-26 15:43:46
## 348                      1.600000      FALSE 2023-02-24 11:44:11
## 356                      2.666667      FALSE 2023-04-21 16:49:20
## 366                      2.600000      FALSE 2023-05-12 10:15:53
## 378                      3.600000      FALSE 2023-05-12 11:42:59
## 387                      2.000000      FALSE 2023-07-07 17:37:10
## 401                      2.000000      FALSE 2023-04-21 16:27:51
## 411                      2.400000      FALSE 2023-04-28 13:51:32
## 414                      1.666667      FALSE 2023-04-14 16:42:51
## 421                      2.200000      FALSE 2023-02-17 11:51:02
## 424                      2.333333      FALSE 2023-06-26 18:59:34
## 425                      3.200000      FALSE 2023-04-14 17:20:04
## 429                      3.333333      FALSE 2023-07-06 17:58:47
## 431                      2.200000      FALSE 2023-05-12 11:47:45
## 433                      2.200000      FALSE 2023-04-07 16:34:58
## 436                      2.200000      FALSE 2023-05-11 14:57:46
## 439                      2.600000      FALSE 2023-07-13 13:02:18
## 457                      3.400000      FALSE 2023-07-14 16:51:09
## 519                      3.000000      FALSE 2023-08-04 16:36:03
## 542                      1.800000      FALSE 2023-03-10 11:53:42
## 547                      4.000000      FALSE 2023-04-28 10:13:44
## 553                      2.250000      FALSE 2023-04-17 17:06:13
## 559                      3.000000      FALSE 2023-08-03 16:36:15
## 570                      2.600000      FALSE 2023-04-17 17:45:31
## 607                      4.000000      FALSE 2023-07-24 17:40:02
## 611                      2.333333      FALSE 2023-07-17 19:39:59
## 615                      2.600000      FALSE 2023-07-07 18:12:21
## 634                      2.000000      FALSE 2023-07-17 19:00:09
## 645                      3.000000      FALSE 2023-04-17 18:48:16
## 649                      1.666667      FALSE 2023-05-12 10:16:04
## 655                      2.000000      FALSE 2023-04-24 17:33:24
## 656                      2.666667      FALSE 2023-03-20 17:06:51
## 666                      3.666667      FALSE 2023-02-22 17:30:45
## 685                      3.400000      FALSE 2023-03-07 21:04:25
## 687                      2.333333      FALSE 2023-06-22 21:37:32
## 688                      2.000000      FALSE 2023-03-07 17:00:26
```

```
## 691                  3.000000      FALSE 2023-04-21 11:01:01
##      Last modified time Final_Accuracy Human Consultancy Sample
## 21  2023-04-28 11:30:24           TRUE                    FALSE
## 43  2023-05-26 10:54:34           TRUE                    FALSE
## 78  2023-05-19 16:20:39           TRUE                    FALSE
## 81  2023-06-23 11:56:33           TRUE                    FALSE
## 91  2023-07-27 16:36:48           TRUE                    FALSE
## 94  2023-06-29 18:41:52           TRUE                    FALSE
## 99  2023-07-31 15:39:55           TRUE                    FALSE
## 113 2023-04-21 16:48:05           TRUE                    FALSE
## 136 2023-07-24 15:45:08           TRUE                    FALSE
## 140 2023-06-12 16:25:09           TRUE                    FALSE
## 149 2023-04-12 17:18:09           TRUE                    FALSE
## 177 2023-04-28 10:25:57           TRUE                    FALSE
## 179 2023-07-20 15:41:51           TRUE                    FALSE
## 185 2023-04-28 16:44:08           TRUE                    FALSE
## 186 2023-05-12 16:09:16           TRUE                    FALSE
## 191 2023-05-19 16:52:53           TRUE                    FALSE
## 202 2023-04-29 18:16:46           TRUE                    FALSE
## 211 2023-05-18 11:38:29           TRUE                    FALSE
## 215 2023-02-13 16:41:56           TRUE                    FALSE
## 216 2023-05-01 16:31:54           TRUE                    FALSE
## 219 2023-07-28 15:39:59           TRUE                    FALSE
## 236 2023-06-26 17:15:36           TRUE                    FALSE
## 240 2023-06-23 23:14:19           TRUE                    FALSE
## 241 2023-07-17 16:33:07           TRUE                    FALSE
## 254 2023-07-17 15:04:00           TRUE                    FALSE
## 270 2023-04-28 16:50:44           TRUE                    FALSE
## 276 2023-05-15 16:10:35           TRUE                    FALSE
## 290 2023-07-06 15:47:04           TRUE                    FALSE
## 306 2023-07-17 18:30:49           TRUE                    FALSE
## 324 2023-05-11 16:49:22           TRUE                    FALSE
## 331 2023-05-11 15:50:12           TRUE                    FALSE
## 332 2023-04-29 17:56:08           TRUE                    FALSE
## 338 2023-06-22 18:58:39           TRUE                    FALSE
## 342 2023-06-26 15:57:14           TRUE                    FALSE
## 348 2023-04-28 16:45:16           TRUE                    FALSE
## 356 2023-04-21 16:49:20           TRUE                    FALSE
## 366 2023-05-12 10:15:53           TRUE                    FALSE
## 378 2023-06-12 16:33:57           TRUE                    FALSE
## 387 2023-07-07 17:37:10           TRUE                    FALSE
## 401 2023-04-21 16:27:51           TRUE                    FALSE
## 411 2023-05-12 10:49:32           TRUE                    FALSE
## 414 2023-06-12 16:48:26           TRUE                    FALSE
## 421 2023-05-15 17:10:36           TRUE                    FALSE
## 424 2023-06-26 18:59:34           TRUE                    FALSE
## 425 2023-04-28 10:10:59           TRUE                    FALSE
## 429 2023-07-06 17:58:47           TRUE                    FALSE
## 431 2023-06-12 16:01:09           TRUE                    FALSE
## 433 2023-04-07 16:34:58           TRUE                    FALSE
## 436 2023-05-11 14:57:46           TRUE                    FALSE
## 439 2023-07-13 13:02:18           TRUE                    FALSE
## 457 2023-07-14 16:51:09           TRUE                    FALSE
## 519 2023-08-04 16:36:03           TRUE                    FALSE
```

```
## 542 2023-04-13 16:46:04          TRUE                      FALSE
## 547 2023-06-12 16:24:31          TRUE                      FALSE
## 553 2023-04-18 13:42:45          TRUE                      FALSE
## 559 2023-08-03 16:36:15          TRUE                      FALSE
## 570 2023-04-29 22:45:31          TRUE                      FALSE
## 607 2023-07-24 17:40:02          TRUE                      FALSE
## 611 2023-07-17 19:39:59          TRUE                      FALSE
## 615 2023-07-07 21:30:24          TRUE                      FALSE
## 634 2023-07-17 19:00:09          TRUE                      FALSE
## 645 2023-04-18 14:26:39          TRUE                      FALSE
## 649 2023-05-12 10:16:04          TRUE                      FALSE
## 655 2023-05-24 16:28:55          TRUE                      FALSE
## 656 2023-04-28 16:39:55          TRUE                      FALSE
## 666 2023-02-22 17:30:45          TRUE                      FALSE
## 685 2023-04-28 17:01:26          TRUE                      FALSE
## 687 2023-06-22 21:37:32          TRUE                      FALSE
## 688 2023-04-28 17:38:19          TRUE                      FALSE
## 691 2023-06-12 16:05:11          TRUE                      FALSE
##     AI Consultancy Sample Human Debate Sample AI Debate Sample Sample
## 21                  FALSE               FALSE            FALSE  FALSE
## 43                  FALSE               FALSE            FALSE  FALSE
## 78                  FALSE               FALSE            FALSE  FALSE
## 81                  FALSE               FALSE            FALSE  FALSE
## 91                  FALSE                TRUE            FALSE   TRUE
## 94                  FALSE               FALSE            FALSE  FALSE
## 99                  FALSE               FALSE            FALSE  FALSE
## 113                 FALSE               FALSE            FALSE  FALSE
## 136                 FALSE               FALSE            FALSE  FALSE
## 140                 FALSE                TRUE            FALSE   TRUE
## 149                 FALSE               FALSE            FALSE  FALSE
## 177                 FALSE               FALSE            FALSE  FALSE
## 179                 FALSE               FALSE            FALSE  FALSE
## 185                 FALSE                TRUE            FALSE   TRUE
## 186                 FALSE               FALSE            FALSE  FALSE
## 191                 FALSE               FALSE            FALSE  FALSE
## 202                 FALSE               FALSE            FALSE  FALSE
## 211                 FALSE                TRUE            FALSE   TRUE
## 215                 FALSE                TRUE            FALSE   TRUE
## 216                 FALSE               FALSE            FALSE  FALSE
## 219                 FALSE                TRUE            FALSE   TRUE
## 236                 FALSE               FALSE            FALSE  FALSE
## 240                 FALSE               FALSE            FALSE  FALSE
## 241                 FALSE               FALSE            FALSE  FALSE
## 254                 FALSE               FALSE            FALSE  FALSE
## 270                 FALSE                TRUE            FALSE   TRUE
## 276                 FALSE                TRUE            FALSE   TRUE
## 290                 FALSE                TRUE            FALSE   TRUE
## 306                 FALSE               FALSE            FALSE  FALSE
## 324                 FALSE                TRUE            FALSE   TRUE
## 331                 FALSE                TRUE            FALSE   TRUE
## 332                 FALSE                TRUE            FALSE   TRUE
## 338                 FALSE                TRUE            FALSE   TRUE
## 342                 FALSE               FALSE            FALSE  FALSE
## 348                 FALSE                TRUE            FALSE   TRUE
```

```
## 356                FALSE                TRUE              FALSE   TRUE
## 366                FALSE               FALSE              FALSE  FALSE
## 378                FALSE                TRUE              FALSE   TRUE
## 387                FALSE               FALSE              FALSE  FALSE
## 401                FALSE               FALSE              FALSE  FALSE
## 411                FALSE               FALSE              FALSE  FALSE
## 414                FALSE                TRUE              FALSE   TRUE
## 421                FALSE                TRUE              FALSE   TRUE
## 424                FALSE                TRUE              FALSE   TRUE
## 425                FALSE                TRUE              FALSE   TRUE
## 429                FALSE               FALSE              FALSE  FALSE
## 431                FALSE                TRUE              FALSE   TRUE
## 433                FALSE                TRUE              FALSE   TRUE
## 436                FALSE                TRUE              FALSE   TRUE
## 439                FALSE                TRUE              FALSE   TRUE
## 457                FALSE                TRUE              FALSE   TRUE
## 519                FALSE                TRUE              FALSE   TRUE
## 542                FALSE                TRUE              FALSE   TRUE
## 547                FALSE                TRUE              FALSE   TRUE
## 553                FALSE                TRUE              FALSE   TRUE
## 559                FALSE                TRUE              FALSE   TRUE
## 570                FALSE                TRUE              FALSE   TRUE
## 607                FALSE                TRUE              FALSE   TRUE
## 611                FALSE                TRUE              FALSE   TRUE
## 615                FALSE                TRUE              FALSE   TRUE
## 634                FALSE                TRUE              FALSE   TRUE
## 645                FALSE                TRUE              FALSE   TRUE
## 649                FALSE                TRUE              FALSE   TRUE
## 655                FALSE                TRUE              FALSE   TRUE
## 656                FALSE                TRUE              FALSE   TRUE
## 666                FALSE                TRUE              FALSE   TRUE
## 685                FALSE                TRUE              FALSE   TRUE
## 687                FALSE                TRUE              FALSE   TRUE
## 688                FALSE                TRUE              FALSE   TRUE
## 691                FALSE                TRUE              FALSE   TRUE
##     Consultancy Sample initial_question_weights
## 21              FALSE                0.5000000
## 43              FALSE                0.5000000
## 78              FALSE                0.5000000
## 81              FALSE                0.2500000
## 91              FALSE                0.1666667
## 94              FALSE                0.5000000
## 99              FALSE                0.2500000
## 113             FALSE                0.3333333
## 136             FALSE                0.1428571
## 140             FALSE                1.0000000
## 149             FALSE                0.2500000
## 177             FALSE                0.5000000
## 179             FALSE                0.5000000
## 185             FALSE                1.0000000
## 186             FALSE                0.5000000
## 191             FALSE                0.2000000
## 202             FALSE                0.5000000
## 211             FALSE                0.5000000
```

```
## 215             FALSE               1.0000000
## 216             FALSE               0.5000000
## 219             FALSE               0.1666667
## 236             FALSE               0.5000000
## 240             FALSE               0.5000000
## 241             FALSE               0.2500000
## 254             FALSE               0.5000000
## 270             FALSE               1.0000000
## 276             FALSE               0.5000000
## 290             FALSE               0.2500000
## 306             FALSE               0.5000000
## 324             FALSE               0.5000000
## 331             FALSE               0.2500000
## 332             FALSE               0.5000000
## 338             FALSE               0.5000000
## 342             FALSE               0.5000000
## 348             FALSE               1.0000000
## 356             FALSE               0.3333333
## 366             FALSE               0.2500000
## 378             FALSE               0.3333333
## 387             FALSE               0.5000000
## 401             FALSE               0.2500000
## 411             FALSE               0.5000000
## 414             FALSE               0.5000000
## 421             FALSE               1.0000000
## 424             FALSE               0.5000000
## 425             FALSE               0.3333333
## 429             FALSE               0.2000000
## 431             FALSE               1.0000000
## 433             FALSE               0.3333333
## 436             FALSE               1.0000000
## 439             FALSE               0.2500000
## 457             FALSE               0.2000000
## 519             FALSE               0.1666667
## 542             FALSE               0.5000000
## 547             FALSE               0.5000000
## 553             FALSE               1.0000000
## 559             FALSE               0.2500000
## 570             FALSE               0.2500000
## 607             FALSE               0.1428571
## 611             FALSE               0.5000000
## 615             FALSE               0.5000000
## 634             FALSE               0.2500000
## 645             FALSE               0.3333333
## 649             FALSE               0.3333333
## 655             FALSE               0.5000000
## 656             FALSE               0.2000000
## 666             FALSE               0.3333333
## 685             FALSE               0.3333333
## 687             FALSE               0.2500000
## 688             FALSE               1.0000000
## 691             FALSE               0.5000000
##     initial_question_weights_grouped_setting
## 21                                       0.5
```

```
## 43                                     0.5
## 78                                     0.5
## 81                                     0.5
## 91                                     1.0
## 94                                     0.5
## 99                                     0.5
## 113                                    0.5
## 136                                    0.5
## 140                                    1.0
## 149                                    0.5
## 177                                    0.5
## 179                                    0.5
## 185                                    1.0
## 186                                    0.5
## 191                                    0.5
## 202                                    0.5
## 211                                    0.5
## 215                                    1.0
## 216                                    0.5
## 219                                    1.0
## 236                                    0.5
## 240                                    0.5
## 241                                    0.5
## 254                                    0.5
## 270                                    1.0
## 276                                    0.5
## 290                                    0.5
## 306                                    0.5
## 324                                    0.5
## 331                                    0.5
## 332                                    0.5
## 338                                    0.5
## 342                                    0.5
## 348                                    1.0
## 356                                    1.0
## 366                                    0.5
## 378                                    0.5
## 387                                    0.5
## 401                                    0.5
## 411                                    0.5
## 414                                    0.5
## 421                                    1.0
## 424                                    0.5
## 425                                    0.5
## 429                                    0.5
## 431                                    1.0
## 433                                    1.0
## 436                                    1.0
## 439                                    0.5
## 457                                    1.0
## 519                                    1.0
## 542                                    1.0
## 547                                    0.5
## 553                                    1.0
```

```
## 559                                       1.0
## 570                                       0.5
## 607                                       0.5
## 611                                       1.0
## 615                                       0.5
## 634                                       0.5
## 645                                       0.5
## 649                                       0.5
## 655                                       0.5
## 656                                       0.5
## 666                                       1.0
## 685                                       1.0
## 687                                       0.5
## 688                                       1.0
## 691                                       0.5
##     sampled_consultancies_all_debates_weights
## 21                                  0.5000000
## 43                                  0.5000000
## 78                                  0.5000000
## 81                                  0.3333333
## 91                                  0.2000000
## 94                                  0.5000000
## 99                                  0.2500000
## 113                                 0.3333333
## 136                                 0.1666667
## 140                                 1.0000000
## 149                                 0.2500000
## 177                                 0.5000000
## 179                                 0.5000000
## 185                                 1.0000000
## 186                                 0.5000000
## 191                                 0.2000000
## 202                                 0.5000000
## 211                                 0.5000000
## 215                                 1.0000000
## 216                                 0.5000000
## 219                                 0.2000000
## 236                                 0.5000000
## 240                                 0.5000000
## 241                                 0.3333333
## 254                                 0.5000000
## 270                                 1.0000000
## 276                                 0.5000000
## 290                                 0.3333333
## 306                                 0.5000000
## 324                                 0.5000000
## 331                                 0.3333333
## 332                                 0.5000000
## 338                                 0.5000000
## 342                                 0.5000000
## 348                                 1.0000000
## 356                                 0.3333333
## 366                                 0.2500000
## 378                                 0.3333333
```

```
## 387                                     0.5000000
## 401                                     0.3333333
## 411                                     0.5000000
## 414                                     0.5000000
## 421                                     1.0000000
## 424                                     0.5000000
## 425                                     0.3333333
## 429                                     0.2000000
## 431                                     1.0000000
## 433                                     0.3333333
## 436                                     1.0000000
## 439                                     0.2500000
## 457                                     0.2500000
## 519                                     0.2000000
## 542                                     0.5000000
## 547                                     0.5000000
## 553                                     1.0000000
## 559                                     0.2500000
## 570                                     0.2500000
## 607                                     0.1666667
## 611                                     0.5000000
## 615                                     0.5000000
## 634                                     0.2500000
## 645                                     0.3333333
## 649                                     0.3333333
## 655                                     0.5000000
## 656                                     0.2000000
## 666                                     0.3333333
## 685                                     0.3333333
## 687                                     0.2500000
## 688                                     1.0000000
## 691                                     0.5000000
##     sampled_consultancies_all_debates_weights_setting
## 21                                                0.5
## 43                                                0.5
## 78                                                0.5
## 81                                                0.5
## 91                                                1.0
## 94                                                0.5
## 99                                                0.5
## 113                                               0.5
## 136                                               0.5
## 140                                               1.0
## 149                                               0.5
## 177                                               0.5
## 179                                               0.5
## 185                                               1.0
## 186                                               0.5
## 191                                               0.5
## 202                                               0.5
## 211                                               0.5
## 215                                               1.0
## 216                                               0.5
## 219                                               1.0
```

```
## 236                                                  0.5
## 240                                                  0.5
## 241                                                  0.5
## 254                                                  0.5
## 270                                                  1.0
## 276                                                  0.5
## 290                                                  0.5
## 306                                                  0.5
## 324                                                  0.5
## 331                                                  0.5
## 332                                                  0.5
## 338                                                  0.5
## 342                                                  0.5
## 348                                                  1.0
## 356                                                  1.0
## 366                                                  0.5
## 378                                                  0.5
## 387                                                  0.5
## 401                                                  0.5
## 411                                                  0.5
## 414                                                  0.5
## 421                                                  1.0
## 424                                                  0.5
## 425                                                  0.5
## 429                                                  0.5
## 431                                                  1.0
## 433                                                  1.0
## 436                                                  1.0
## 439                                                  0.5
## 457                                                  1.0
## 519                                                  1.0
## 542                                                  1.0
## 547                                                  0.5
## 553                                                  1.0
## 559                                                  1.0
## 570                                                  0.5
## 607                                                  0.5
## 611                                                  1.0
## 615                                                  0.5
## 634                                                  0.5
## 645                                                  0.5
## 649                                                  0.5
## 655                                                  0.5
## 656                                                  0.5
## 666                                                  1.0
## 685                                                  1.0
## 687                                                  0.5
## 688                                                  1.0
## 691                                                  0.5
##     sampled_consultancies_all_debates_weights_grouped_setting
## 21                                                          0.5
## 43                                                          0.5
## 78                                                          0.5
## 81                                                          0.5
```

```
## 91                                                        1.0
## 94                                                        0.5
## 99                                                        0.5
## 113                                                       0.5
## 136                                                       0.5
## 140                                                       1.0
## 149                                                       0.5
## 177                                                       0.5
## 179                                                       0.5
## 185                                                       1.0
## 186                                                       0.5
## 191                                                       0.5
## 202                                                       0.5
## 211                                                       0.5
## 215                                                       1.0
## 216                                                       0.5
## 219                                                       1.0
## 236                                                       0.5
## 240                                                       0.5
## 241                                                       0.5
## 254                                                       0.5
## 270                                                       1.0
## 276                                                       0.5
## 290                                                       0.5
## 306                                                       0.5
## 324                                                       0.5
## 331                                                       0.5
## 332                                                       0.5
## 338                                                       0.5
## 342                                                       0.5
## 348                                                       1.0
## 356                                                       1.0
## 366                                                       0.5
## 378                                                       0.5
## 387                                                       0.5
## 401                                                       0.5
## 411                                                       0.5
## 414                                                       0.5
## 421                                                       1.0
## 424                                                       0.5
## 425                                                       0.5
## 429                                                       0.5
## 431                                                       1.0
## 433                                                       1.0
## 436                                                       1.0
## 439                                                       0.5
## 457                                                       1.0
## 519                                                       1.0
## 542                                                       1.0
## 547                                                       0.5
## 553                                                       1.0
## 559                                                       1.0
## 570                                                       0.5
## 607                                                       0.5
```

```
## 611                                                    1.0
## 615                                                    0.5
## 634                                                    0.5
## 645                                                    0.5
## 649                                                    0.5
## 655                                                    0.5
## 656                                                    0.5
## 666                                                    1.0
## 685                                                    1.0
## 687                                                    0.5
## 688                                                    1.0
## 691                                                    0.5
##     sampled_consultancies_debates_weights
## 21                              0.0000000
## 43                              0.0000000
## 78                              0.0000000
## 81                              0.0000000
## 91                              0.2500000
## 94                              0.0000000
## 99                              0.0000000
## 113                             0.0000000
## 136                             0.0000000
## 140                             1.0000000
## 149                             0.0000000
## 177                             0.0000000
## 179                             0.0000000
## 185                             1.0000000
## 186                             0.0000000
## 191                             0.0000000
## 202                             0.0000000
## 211                             1.0000000
## 215                             1.0000000
## 216                             0.0000000
## 219                             0.2500000
## 236                             0.0000000
## 240                             0.0000000
## 241                             0.0000000
## 254                             0.0000000
## 270                             1.0000000
## 276                             1.0000000
## 290                             0.5000000
## 306                             0.0000000
## 324                             1.0000000
## 331                             0.5000000
## 332                             1.0000000
## 338                             1.0000000
## 342                             0.0000000
## 348                             1.0000000
## 356                             0.3333333
## 366                             0.0000000
## 378                             0.5000000
## 387                             0.0000000
## 401                             0.0000000
## 411                             0.0000000
```

```
## 414                            1.0000000
## 421                            1.0000000
## 424                            1.0000000
## 425                            0.5000000
## 429                            0.0000000
## 431                            1.0000000
## 433                            0.3333333
## 436                            1.0000000
## 439                            0.3333333
## 457                            0.2500000
## 519                            0.2500000
## 542                            0.5000000
## 547                            1.0000000
## 553                            1.0000000
## 559                            0.2500000
## 570                            0.3333333
## 607                            0.2500000
## 611                            0.5000000
## 615                            1.0000000
## 634                            0.3333333
## 645                            0.5000000
## 649                            0.5000000
## 655                            1.0000000
## 656                            0.2500000
## 666                            0.3333333
## 685                            0.3333333
## 687                            0.3333333
## 688                            1.0000000
## 691                            1.0000000
##     sampled_consultancies_debates_weights_setting
## 21                                              0
## 43                                              0
## 78                                              0
## 81                                              0
## 91                                              1
## 94                                              0
## 99                                              0
## 113                                             0
## 136                                             0
## 140                                             1
## 149                                             0
## 177                                             0
## 179                                             0
## 185                                             1
## 186                                             0
## 191                                             0
## 202                                             0
## 211                                             1
## 215                                             1
## 216                                             0
## 219                                             1
## 236                                             0
## 240                                             0
## 241                                             0
```

```
## 254                                             0
## 270                                             1
## 276                                             1
## 290                                             1
## 306                                             0
## 324                                             1
## 331                                             1
## 332                                             1
## 338                                             1
## 342                                             0
## 348                                             1
## 356                                             1
## 366                                             0
## 378                                             1
## 387                                             0
## 401                                             0
## 411                                             0
## 414                                             1
## 421                                             1
## 424                                             1
## 425                                             1
## 429                                             0
## 431                                             1
## 433                                             1
## 436                                             1
## 439                                             1
## 457                                             1
## 519                                             1
## 542                                             1
## 547                                             1
## 553                                             1
## 559                                             1
## 570                                             1
## 607                                             1
## 611                                             1
## 615                                             1
## 634                                             1
## 645                                             1
## 649                                             1
## 655                                             1
## 656                                             1
## 666                                             1
## 685                                             1
## 687                                             1
## 688                                             1
## 691                                             1
##     sampled_consultancies_debates_weights_grouped_setting
## 21                                                      0
## 43                                                      0
## 78                                                      0
## 81                                                      0
## 91                                                      1
## 94                                                      0
## 99                                                      0
```

```
## 113                                                        0
## 136                                                        0
## 140                                                        1
## 149                                                        0
## 177                                                        0
## 179                                                        0
## 185                                                        1
## 186                                                        0
## 191                                                        0
## 202                                                        0
## 211                                                        1
## 215                                                        1
## 216                                                        0
## 219                                                        1
## 236                                                        0
## 240                                                        0
## 241                                                        0
## 254                                                        0
## 270                                                        1
## 276                                                        1
## 290                                                        1
## 306                                                        0
## 324                                                        1
## 331                                                        1
## 332                                                        1
## 338                                                        1
## 342                                                        0
## 348                                                        1
## 356                                                        1
## 366                                                        0
## 378                                                        1
## 387                                                        0
## 401                                                        0
## 411                                                        0
## 414                                                        1
## 421                                                        1
## 424                                                        1
## 425                                                        1
## 429                                                        0
## 431                                                        1
## 433                                                        1
## 436                                                        1
## 439                                                        1
## 457                                                        1
## 519                                                        1
## 542                                                        1
## 547                                                        1
## 553                                                        1
## 559                                                        1
## 570                                                        1
## 607                                                        1
## 611                                                        1
## 615                                                        1
## 634                                                        1
```

```
## 645                                                    1
## 649                                                    1
## 655                                                    1
## 656                                                    1
## 666                                                    1
## 685                                                    1
## 687                                                    1
## 688                                                    1
## 691                                                    1
##                                                check Reward penalty 0.5  fpc
## 21                   Adelle Fernandomonopoly-1          -2.5145732 0.70
## 43            Adelle Fernandotollivers-orbit-1          -1.1520031 0.90
## 78                         Aliyaah Toussaintrx-3          -0.5144996 0.99
## 81        Aliyaah Toussaintstranger-from-space-0          -2.0144996 0.99
## 91  Aliyaah Toussaintthe-long-remembered-thunder-1          -1.5291463 0.98
## 94 Aliyaah Toussaintthe-princess-and-the-physicist-4          -2.0144996 0.99
## 99        Aliyaah Toussaintthe-starsent-knaves-2          -2.2344653 0.85
## 113                         Anuj Jaincosmic-yoyo-0          -2.0144996 0.99
## 136               Anuj Jainout-of-the-iron-womb-0          -2.0144996 0.99
## 140                    Anuj Jainplanet-of-dread-2          -1.0144996 0.99
## 149              Anuj Jainthe-air-of-castor-oil-5          -1.7344653 0.85
## 177                          David Reinmonopoly-2          -1.7344653 0.85
## 179             David Reinpeggy-finds-the-theatre-4          -2.1520031 0.90
## 185               David Reinstalemate-in-space-0          -1.0144996 0.99
## 186              David Reinstranger-from-space-4          -2.0740006 0.95
## 191            David Reinthe-great-nebraska-sea-1          -1.5740006 0.95
## 202                      Ethan Rosencosmic-yoyo-3          -1.1520031 0.90
## 211             Ethan Rosenstranger-from-space-5          -1.0740006 0.95
## 215             Ethan Rosenthe-man-who-was-six-1          -1.3219281 0.80
## 216               Ethan Rosenthe-monster-maker-4          -1.0144996 0.99
## 219  Jackson Pettyatom-mystery-young-atom-detective-0          -3.3219281 0.80
## 236                        Jackson Pettymuck-man-5          -3.5144996 0.99
## 240                              Jackson Pettyrx-4          -1.6520031 0.90
## 241               Jackson Pettysilence-isdeadly-3          -1.5144996 0.99
## 254     Jackson Pettythe-princess-and-the-physicist-0          -2.0740006 0.95
## 270                 Jessica Lidoctor-universe-0          -1.5145732 0.70
## 276             Jessica Lihow-to-make-friends-11          -1.0144996 0.99
## 290                  Jessica Lisilence-isdeadly-2          -0.5144996 0.99
## 306       Jessica Lithe-princess-and-the-physicist-2          -1.0144996 0.99
## 324                     Julian Michaelmonopoly-0          -1.5144996 0.99
## 331           Julian Michaelstranger-from-space-1          -1.0144996 0.99
## 332                 Julian Michaelsurvival-type-4          -1.0144996 0.99
## 338            Julian Michaelthe-monster-maker-3          -1.5144996 0.99
## 342       Julian Michaelthe-spicy-sound-of-success-4          -2.0144996 0.99
## 348           Julien Diranimanners-and-customs-1          -1.7344653 0.85
## 356           Noor Mirza-Rashiddoctor-universe-5          -2.2344653 0.85
## 366                    Noor Mirza-Rashidvolpla-2          -1.5740006 0.95
## 378             Reeya Kansrahow-to-make-friends-0          -1.5291463 0.98
## 387                      Reeya Kansramuck-man-7          -2.1844246 0.88
## 401             Reeya Kansrathe-monster-maker-1          -1.0588937 0.96
## 411                Salsabila Mahdibreak-a-leg-5          -1.0144996 0.99
## 414               Salsabila Mahdicosmic-yoyo-2          -1.0144996 0.99
## 421         Salsabila Mahdimanners-and-customs-0          -1.5144996 0.99
## 424                   Salsabila Mahdimuck-man-4          -1.5144996 0.99
```

```
## 425                Salsabila Mahdiplanet-of-dread-1         -1.5144996 0.99
## 429               Salsabila Mahdisilence-isdeadly-6         -2.0144996 0.99
## 431            Salsabila Mahdistranger-from-space-2         -1.0144996 0.99
## 433             Salsabila Mahdithe-happy-castaway-2         -1.5144996 0.99
## 436           Salsabila Mahdithe-reluctant-heroes-2         -2.0144996 0.99
## 439            Salsabila Mahdithe-starsent-knaves-0         -3.0740006 0.95
## 457                     Sam Jincoming-of-the-gods-2         -1.5144996 0.99
## 519                  Sam Jinvenus-is-a-mans-world-0         -1.5144996 0.99
## 542                  Sean Wanglost-in-translation-3         -1.0291463 0.98
## 547              Sean Wangpeggy-finds-the-theatre-0         -1.1520031 0.90
## 553                        Sean Wangsurvival-type-0         -0.5291463 0.98
## 559                         Sean Wangthe-cool-war-0         -1.5144996 0.99
## 570                               Sean Wangvolpla-3         -1.0740006 0.95
## 607             Shlomo Kofmanout-of-the-iron-womb-1         -0.5892673 0.94
## 611               Shlomo Kofmanpied-piper-of-mars-8         -2.1360615 0.91
## 615                               Shlomo Kofmanrx-5         -2.2175914 0.86
## 634                  Shlomo Kofmanthe-starbusters-3         -1.5439433 0.97
## 645                      Shreeram Modicosmic-yoyo-1         -2.0740006 0.95
## 649                    Shreeram Modiin-the-garden-6         -1.0144996 0.99
## 655          Shreeram Modipeggy-finds-the-theatre-2         -0.5144996 0.99
## 656         Shreeram Modiphone-me-in-central-park-5         -1.0144996 0.99
## 666              Shreeram Modithe-man-who-was-six-5         -1.5144996 0.99
## 685          Vishakh Padmakumarstalemate-in-space-2         -1.8219281 0.80
## 687       Vishakh Padmakumarthe-air-of-castor-oil-4         -1.4150375 0.75
## 688    Vishakh Padmakumarthe-desert-and-the-stars-2         -1.9150375 0.75
## 691          Vishakh Padmakumarthe-monster-maker-5         -2.8219281 0.80
##        confidence_label color_value
## 21              Neutral -0.71457317
## 43              Neutral -0.25200309
## 78  Confidently Correct -0.06449957
## 81  Confidently Correct -0.21449957
## 91  Confidently Correct -0.17914635
## 94  Confidently Correct -0.21449957
## 99              Neutral -0.43446525
## 113 Confidently Correct -0.21449957
## 136 Confidently Correct -0.21449957
## 140 Confidently Correct -0.11449957
## 149             Neutral -0.38446525
## 177             Neutral -0.38446525
## 179             Neutral -0.35200309
## 185 Confidently Correct -0.11449957
## 186             Neutral -0.27400058
## 191             Neutral -0.22400058
## 202             Neutral -0.25200309
## 211             Neutral -0.17400058
## 215             Neutral -0.42192809
## 216 Confidently Correct -0.11449957
## 219             Neutral -0.62192809
## 236 Confidently Correct -0.36449957
## 240             Neutral -0.30200309
## 241 Confidently Correct -0.16449957
## 254             Neutral -0.27400058
## 270             Neutral -0.61457317
## 276 Confidently Correct -0.11449957
```

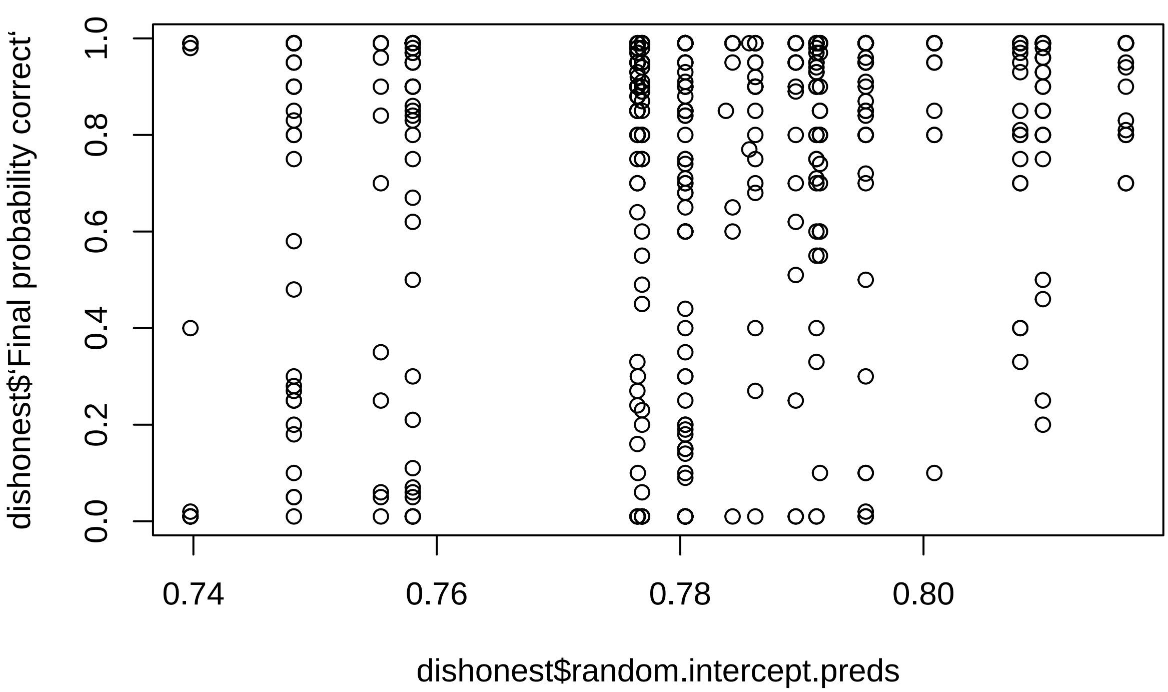
```
## 290 Confidently Correct -0.06449957
## 306 Confidently Correct -0.11449957
## 324 Confidently Correct -0.16449957
## 331 Confidently Correct -0.11449957
## 332 Confidently Correct -0.11449957
## 338 Confidently Correct -0.16449957
## 342 Confidently Correct -0.21449957
## 348             Neutral -0.38446525
## 356             Neutral -0.43446525
## 366             Neutral -0.22400058
## 378 Confidently Correct -0.17914635
## 387             Neutral -0.38442457
## 401 Confidently Correct -0.15889369
## 411 Confidently Correct -0.11449957
## 414 Confidently Correct -0.11449957
## 421 Confidently Correct -0.16449957
## 424 Confidently Correct -0.16449957
## 425 Confidently Correct -0.16449957
## 429 Confidently Correct -0.21449957
## 431 Confidently Correct -0.11449957
## 433 Confidently Correct -0.16449957
## 436 Confidently Correct -0.21449957
## 439             Neutral -0.37400058
## 457 Confidently Correct -0.16449957
## 519 Confidently Correct -0.16449957
## 542 Confidently Correct -0.12914635
## 547             Neutral -0.25200309
## 553 Confidently Correct -0.07914635
## 559 Confidently Correct -0.16449957
## 570             Neutral -0.17400058
## 607             Neutral -0.13926734
## 611             Neutral -0.33606155
## 615             Neutral -0.41759144
## 634 Confidently Correct -0.19394335
## 645             Neutral -0.27400058
## 649 Confidently Correct -0.11449957
## 655 Confidently Correct -0.06449957
## 656 Confidently Correct -0.11449957
## 666 Confidently Correct -0.16449957
## 685             Neutral -0.47192809
## 687             Neutral -0.51503750
## 688             Neutral -0.56503750
## 691             Neutral -0.57192809
```

```r
# Fit the random intercept model and only remove missing values for 'Dishonest debater'
random_intercept_model <- lmer(`Final probability correct` ~ (1|`Dishonest debater`),
                               data = dishonest,
                               REML = TRUE)

# Summary of the model
summary(random_intercept_model)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
```

```
## Formula: ‘Final probability correct‘ ~ (1 | ‘Dishonest debater‘)
##    Data: dishonest
##
## REML criterion at convergence: 301.5
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.5333 -0.1808  0.4992  0.6576  0.8073
##
## Random effects:
##  Groups            Name        Variance Std.Dev.
##  Dishonest debater (Intercept) 0.001508 0.03883
##  Residual                      0.096080 0.30997
## Number of obs: 584, groups:  Dishonest debater, 20
##
## Fixed effects:
##             Estimate Std. Error      df t value       Pr(>|t|)
## (Intercept)  0.78272    0.01658 7.18181    47.2 0.000000000322 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
dishonest$random.intercept.preds = predict(random_intercept_model)
plot(dishonest$random.intercept.preds, dishonest$`Final probability correct`)
```



Debater "Experience", ratings - how many wins?

AI vs Humans

Old vs New

possibly unnessary

Finally, these are how many we get correct in each setting

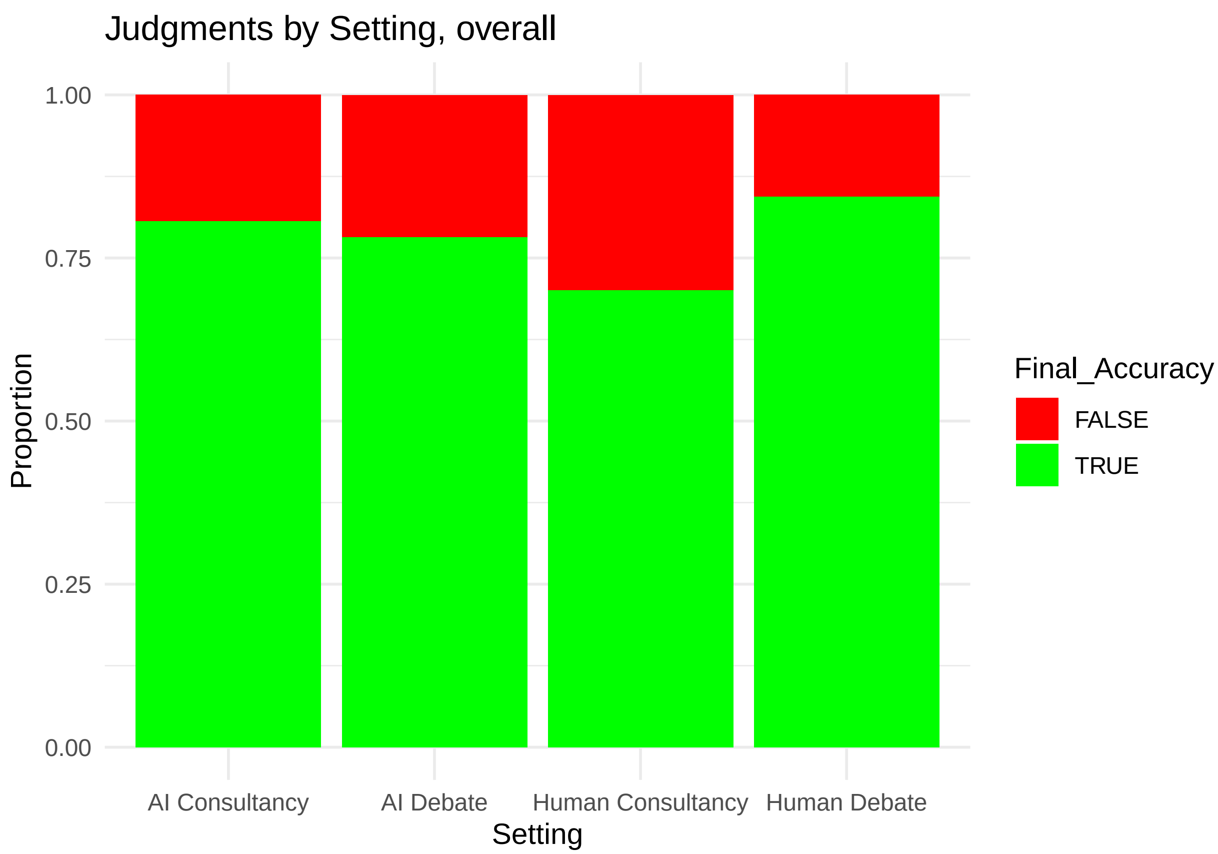
```
judgments_online <- py$judgments_online
table(judgments_online$Final_Accuracy, judgments_online$Final_Setting)
```

```
##
##         AI Consultancy AI Debate Human Consultancy Human Debate
##   FALSE             18        19                32           24
##   TRUE              75        68                75          130
```

```
table(judgments_online$Final_Accuracy, judgments_online$Setting)
```

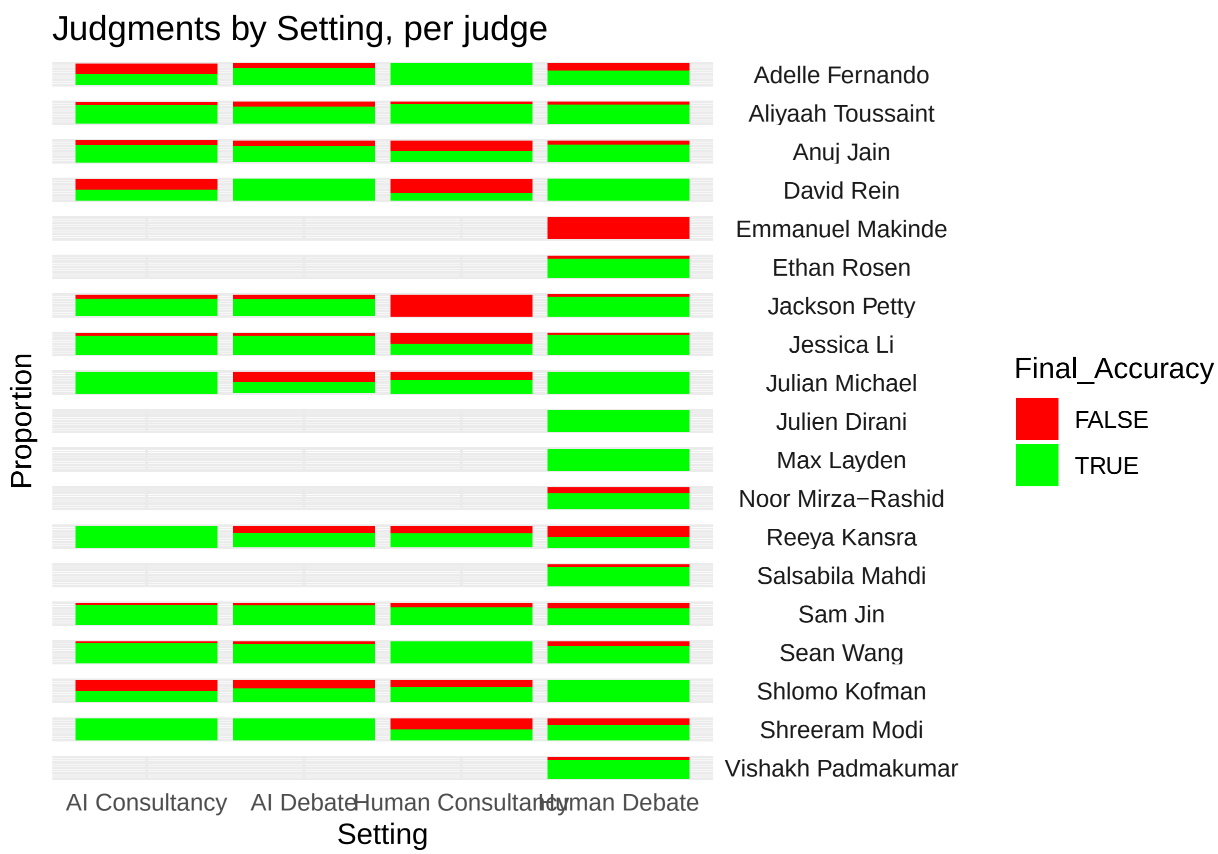
```
##
##         AI Consultancy Dishonest AI Consultancy Honest AI Debate
##   FALSE                        5                    13        19
##   TRUE                        33                    42        68
##
##         Human Consultancy Dishonest Human Consultancy Honest Human Debate
##   FALSE                           26                       6           24
##   TRUE                            33                      42          130
```

```
ggplot(judgments_online, aes(x = Final_Setting, fill = Final_Accuracy)) +
  geom_bar(position = "fill") +
  scale_fill_manual(values = c("TRUE" = "green", "FALSE" = "red")) +
  labs(title = "Judgments by Setting, overall", x = "Setting", y = "Proportion", fill = "Final_Accuracy
  theme_minimal() +
  theme(axis.text.x = element_text())
```



Sneak peak of accuracy differences between judges, but we won't get to that again until models

```
ggplot(judgments_online, aes(x = Final_Setting, fill = Final_Accuracy)) +
  geom_bar(position = "fill") +
  scale_fill_manual(values = c("TRUE" = "green", "FALSE" = "red")) +
  labs(title = "Judgments by Setting, per judge", x = "Setting", y = "Proportion", fill = "Final_Accura
  theme_minimal() +
  theme(axis.text.x = element_text(),#angle = 90, hjust = 1),
        axis.text.y = element_blank(),
        strip.text.y.right = element_text(angle = 0)) +
  facet_grid(rows = "Participant")
```


Judgments by Setting, per judge
Proportion
Adelle Fernando
Aliyaah Toussaint
Anuj Jain
David Rein
Emmanuel Makinde
Ethan Rosen
Jackson Petty
Jessica Li
Julian Michael
Julien Dirani
Max Layden
Noor Mirza−Rashid
Reeya Kansra
Salsabila Mahdi
Sam Jin
Sean Wang
Shlomo Kofman
Shreeram Modi
Vishakh Padmakumar
AI Consultancy
AI Debate
Human Consultancy
Human Debate
Setting
Final_Accuracy
FALSE
TRUE