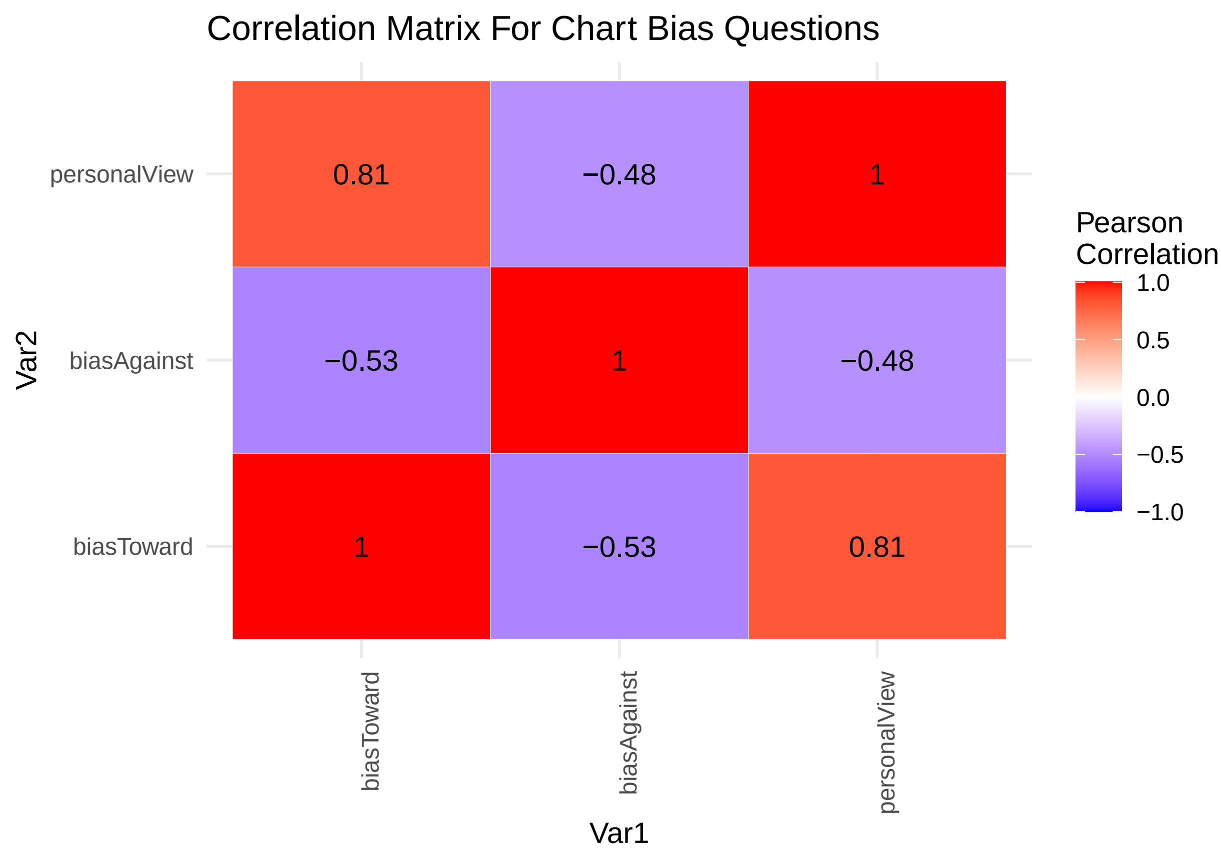
study 2 analysis

2023-03-08

```
## ## FSA v0.9.4. See citation('FSA') if used in publication.
## ## Run fishR() for related website and fishR('IFAR') for related book.

## Loading required package: Matrix

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## Registered S3 methods overwritten by 'car':
##   method       from
##   hist.boot    FSA
##   confint.boot FSA
##
##
## Attaching package: 'rstatix'
##
##
## The following object is masked from 'package:stats':
##
##     filter

##  [1] "pt"                 "winning_party"      "winning_party.1"
##  [4] "winning_scale_1"    "winning_scale_1.1" "pct.chances_blue"
##  [7] "pct.chances_green" "pct_tie"            "biasToward_1"
## [10] "biasToward_1.1"    "biasAgainst_1"     "biasAgainst_1.1"
## [13] "personalView_1"    "personalView_1.1"  "age"
## [16] "education"         "survey_about"      "comments"
## [19] "Order"             "Prime"             "Strength"
## [22] "ChartType"         "Slant"             "Stimulus"
```



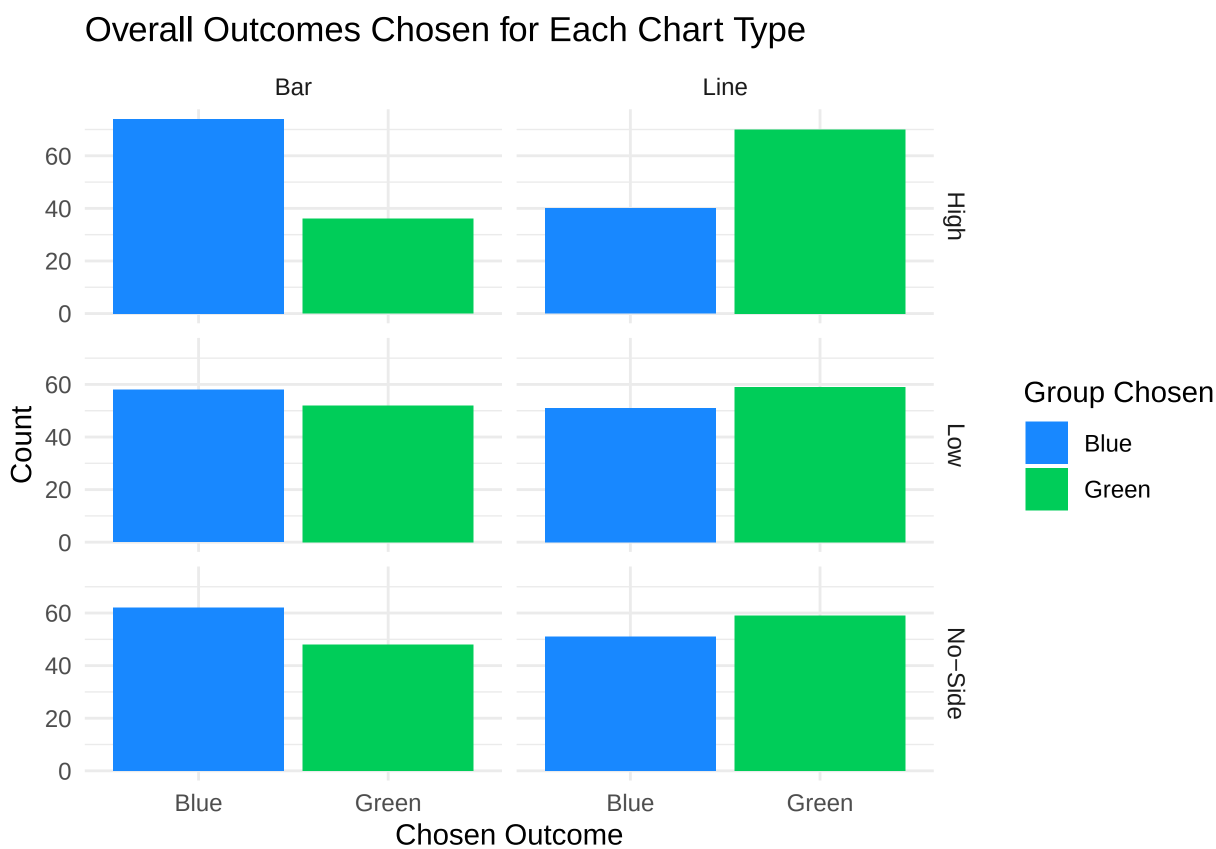

```
## `summarise()` has grouped output by 'full_condition'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 12 x 3
## # Groups:   full_condition [6]
##    full_condition Exclude     n
##    <chr>            <dbl> <int>
##  1 Bar High             0   110
##  2 Bar High             1    15
##  3 Bar Low              0   110
##  4 Bar Low              1     8
##  5 Bar No-Side          0   110
##  6 Bar No-Side          1    12
##  7 Line High            0   110
##  8 Line High            1     6
##  9 Line Low             0   110
## 10 Line Low             1    10
## 11 Line No-Side         0   110
## 12 Line No-Side         1     7
```

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

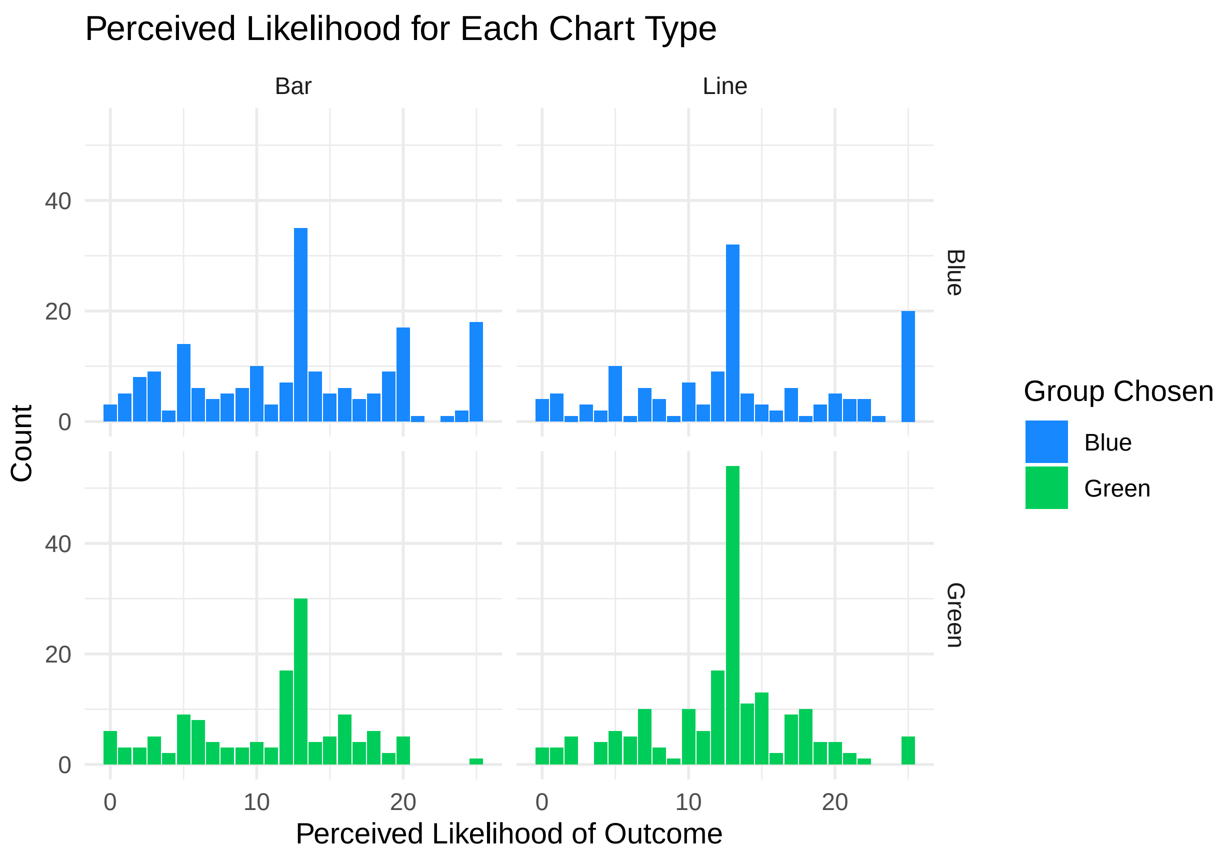
```
## # A tibble: 4 x 3
## # Groups:   ChartType [2]
```

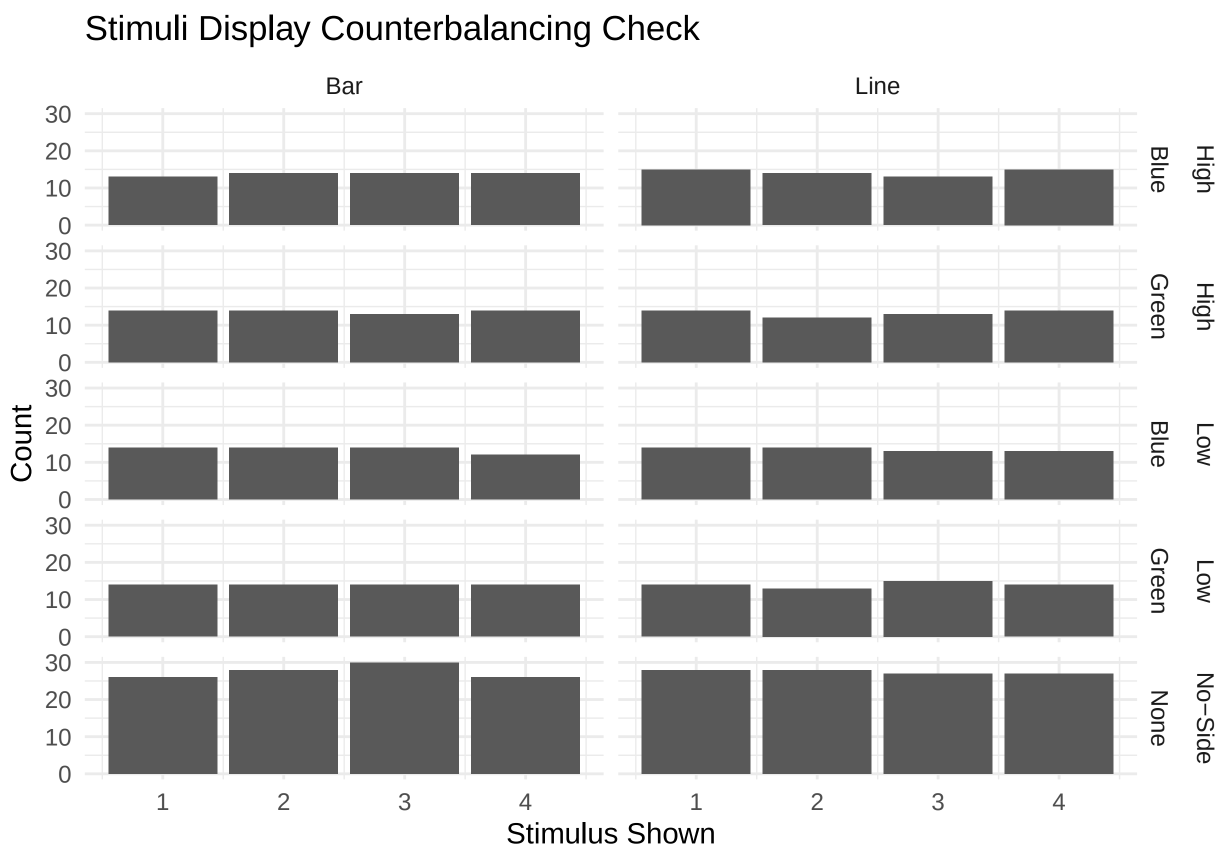
```
##   ChartType Exclude     n
##   <chr>       <dbl> <int>
## 1 Bar             0   330
## 2 Bar             1    35
## 3 Line            0   330
## 4 Line            1    23
```

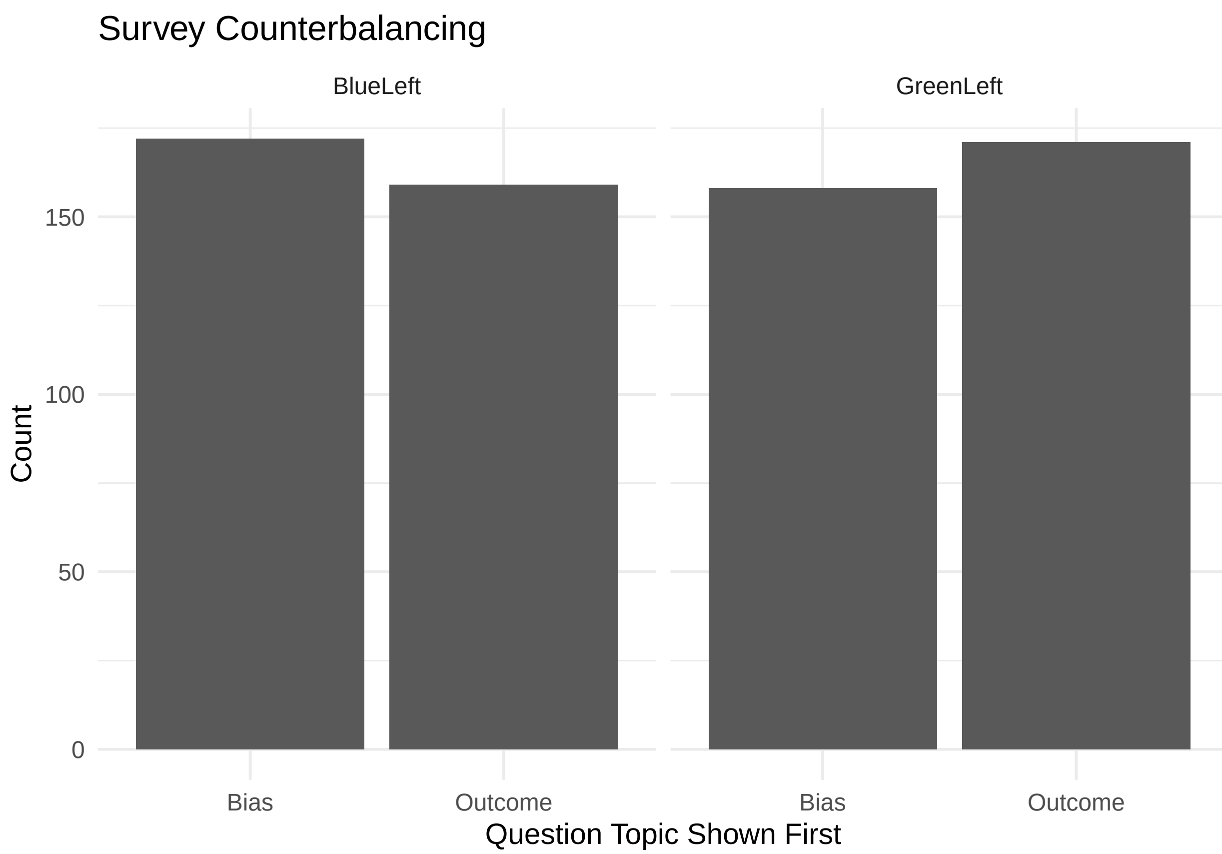



```
## 'summarise()' has grouped output by 'ChartType'. You can override using the
## '.groups' argument.
```

```
## # A tibble: 4 x 3
## # Groups:   ChartType [2]
##   ChartType chosen_outcome     n
##   <chr>     <chr>          <int>
## 1 Bar       Blue             194
## 2 Bar       Green            136
## 3 Line      Blue             142
## 4 Line      Green            188
```


Perceived Likelihood for Each Chart Type
Bar
Line
Blue
Green
Count
40
20
0
Group Chosen
Blue
Green
0
10
20
Perceived Likelihood of Outcome


Stimuli Display Counterbalancing Check
Bar
Line
High
Blue
High
Green
Low
Blue
Low
Green
No−Side
None
Count
30
20
10
0
1
2
3
4
Stimulus Shown



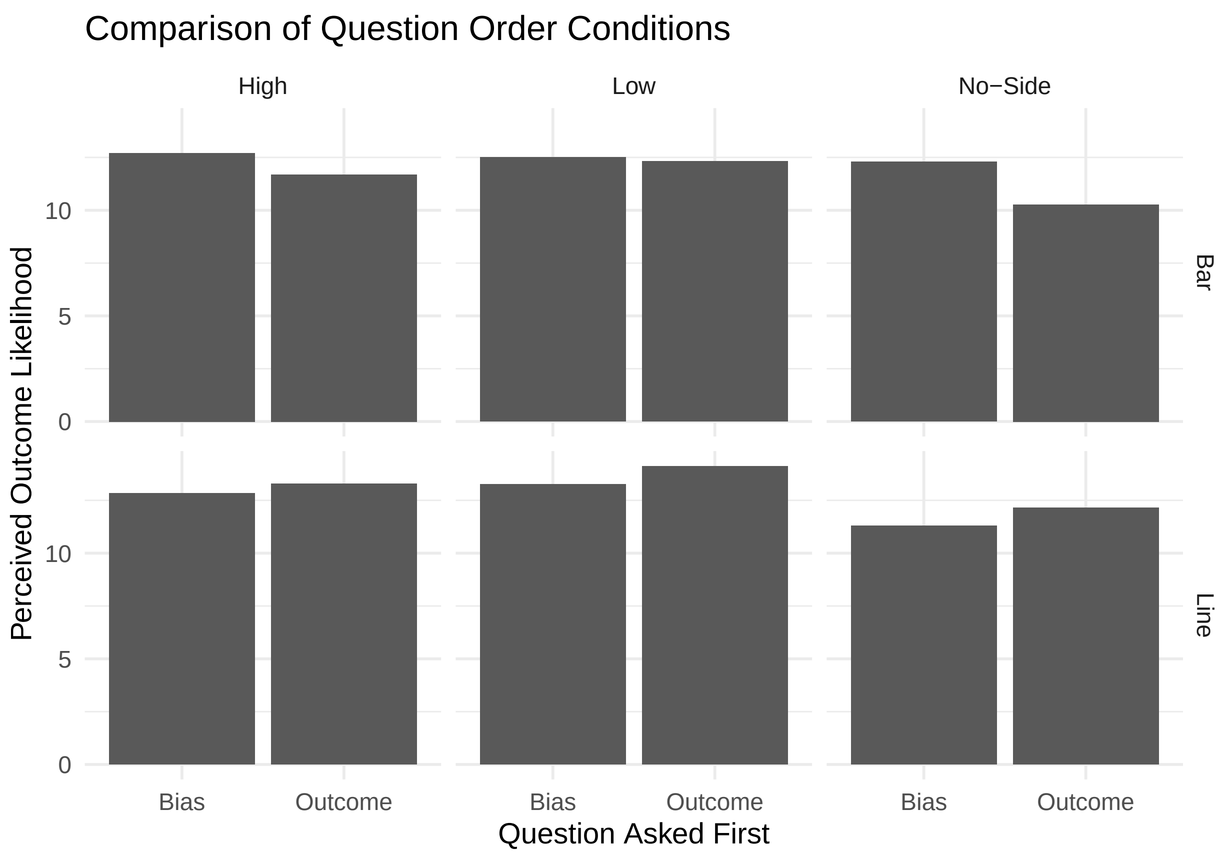
A few checks need to occur before we move on to analysis. First, a broad check for normality.

We will use non-parametric testing with this dataset.

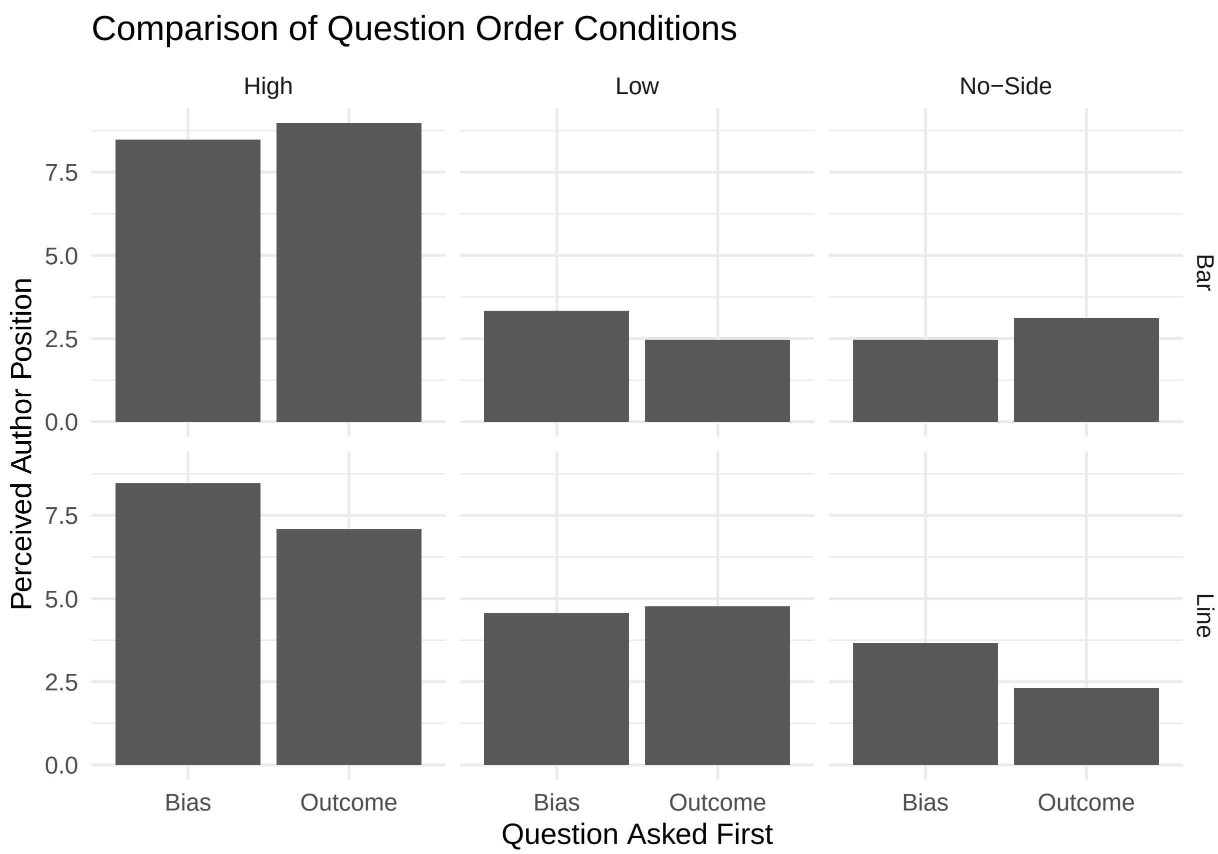
Analysis

First, we want to evaluate whether the question order resulted in different outcome or bias ratings. if this is the case, we need to perform separate analyses for these results, particularly if there is an interaction between priming and bias condition.

```
ggplot(df, aes(y = outcome_confidence_abs, x = Prime))+
  geom_bar(stat = "summary", fun = "mean")+
  facet_grid(ChartType ~ Strength)+
  labs(title = "Comparison of Question Order Conditions",
       y = "Perceived Outcome Likelihood",
       x = "Question Asked First")
```




```r
ggplot(df, aes(y = author_confidence_abs, x = Prime))+
  geom_bar(stat = "summary", fun = "mean")+
  facet_grid(ChartType ~ Strength)+
  labs(title = "Comparison of Question Order Conditions",
       y = "Perceived Author Position",
       x = "Question Asked First")
```

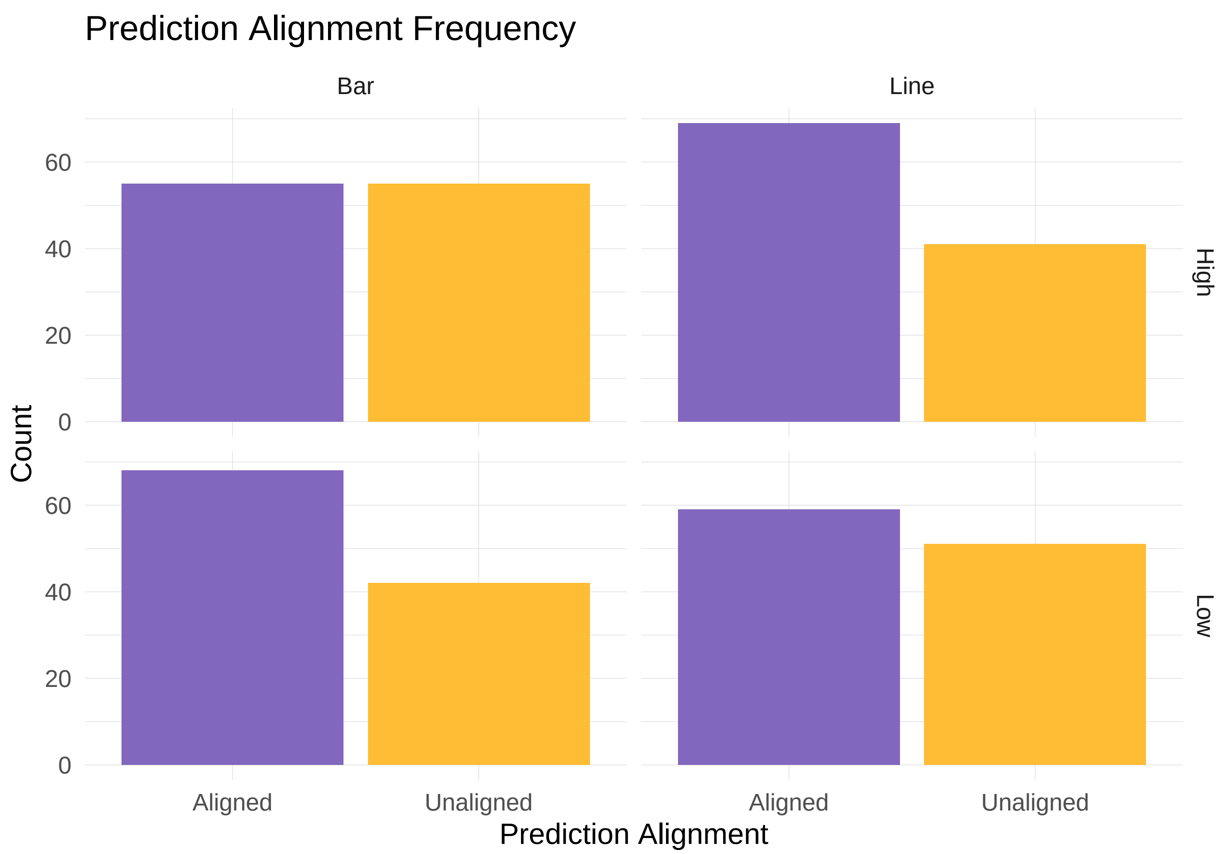


Hypotheses 1.1-1.2

1.1. It is more likely to report judgments in alignment with the group supported in the text. 1.2. Responses aligned with the group supported in the text are more confident than responses unaligned with the group supported and control responses.

```
ggplot(subset(df, treatment == "Treatment"), aes(x = outcome_aligned, y = after_stat(count), fill = outc
  geom_bar()+
  scale_fill_manual(values = c(a, u, c))+
  labs(
    title = "Prediction Alignment Frequency",
    y = "Count",
    x = "Prediction Alignment",
    fill = "Response Alignment"
  )+
  facet_grid(Strength ~ ChartType)+
  theme(legend.position = 'none',
  panel.grid.major = element_line(color = "grey85", size = 0.1),  # Lighten and thin major gridlines
  panel.grid.minor = element_line(color = "grey85", size = 0.05))  # Lighten and thin minor gridlines
```

```
## Warning: The `size` argument of `element_line()` is deprecated as of ggplot2 3.4.0.
## i Please use the `linewidth` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
df %>%
  group_by(ChartType, outcome_aligned) %>%
  summarise(
    n = n()
  )
```

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

```
## # A tibble: 6 x 3
## # Groups:   ChartType [2]
##   ChartType outcome_aligned     n
##   <chr>     <chr>           <int>
## 1 Bar       Aligned           123
## 2 Bar       No-Side           110
## 3 Bar       Unaligned          97
## 4 Line      Aligned           128
## 5 Line      No-Side           110
## 6 Line      Unaligned          92
```

```
t.outcome.bar <- table(subset(bar, treatment == "Treatment")$outcome_aligned)
chisq.test(t.outcome.bar)
```

```
##
```

```
##  Chi-squared test for given probabilities
##
## data:  t.outcome.bar
## X-squared = 3.0727, df = 1, p-value = 0.07962
```

```r
p.outcome.bar <- t.outcome.bar / sum(t.outcome.bar)
dimnames(p.outcome.bar) <- NULL
ES.h(p.outcome.bar[1], p.outcome.bar[2])
```

```
## [1] 0.2369173
```

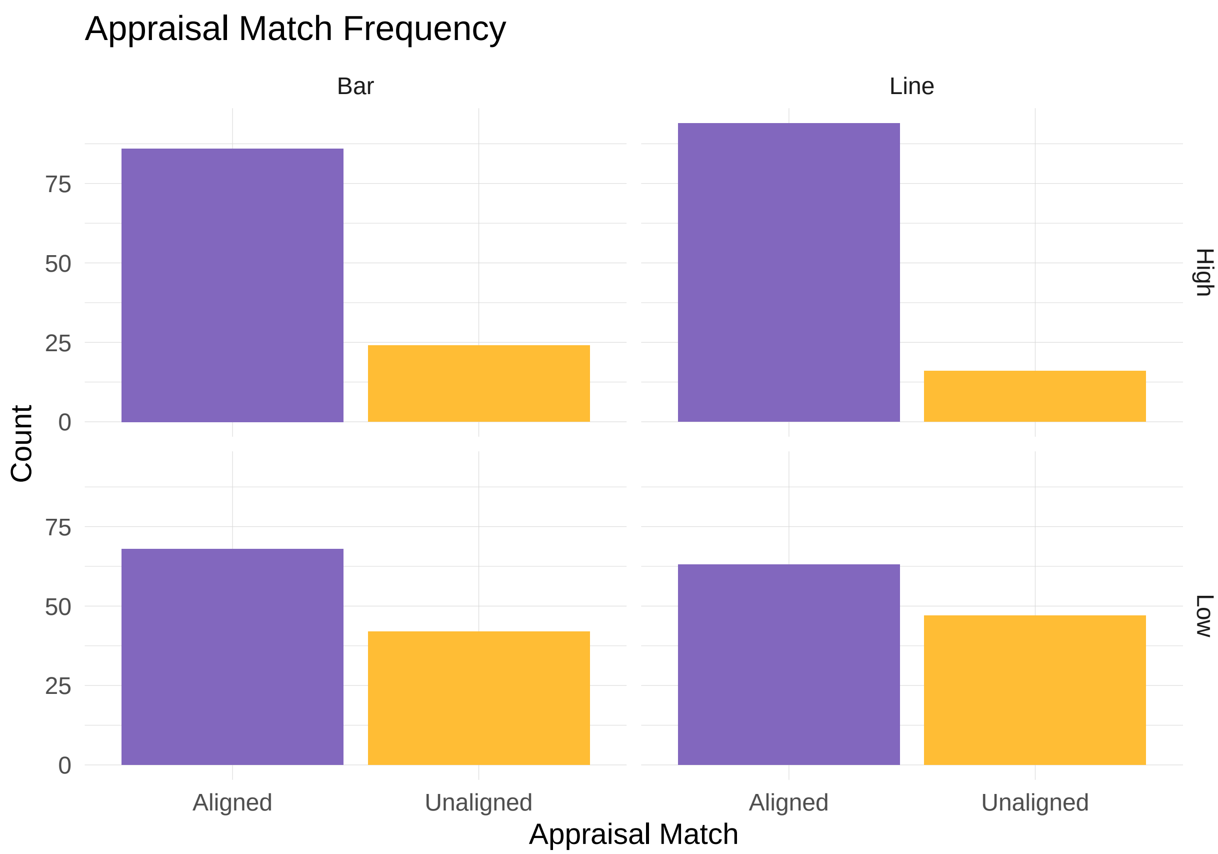
```r
t.outcome.line <- table(subset(line, treatment == "Treatment" & Strength == "High")$outcome_aligned)
chisq.test(t.outcome.line)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  t.outcome.line
## X-squared = 7.1273, df = 1, p-value = 0.007592
```

```r
p.outcome.line <- t.outcome.line / sum(t.outcome.line)
dimnames(p.outcome.line) <- NULL
ES.h(p.outcome.line[1], p.outcome.line[2])
```

```
## [1] 0.5147553
```

```r
ggplot(subset(df, treatment == "Treatment"), aes(x = author_aligned, y = after_stat(count), fill = auth
  geom_bar()+
  scale_fill_manual(values = c(a, u, c), labels= c("Matched", "Unmatched"))+
  labs(
    title = "Appraisal Match Frequency",
    y = "Count",
    x = "Appraisal Match"
  )+
  facet_grid(Strength ~ChartType)+
  theme(legend.position = 'none',
  panel.grid.major = element_line(color = "grey85", size = 0.1),  # Lighten and thin major gridlines
  panel.grid.minor = element_line(color = "grey85", size = 0.05))  # Lighten and thin minor gridlines
```




```r
t.author.bar <- table(subset(bar, treatment == "Treatment" & Strength == "High")$author_aligned)
chisq.test(t.author.bar)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  t.author.bar
## X-squared = 34.945, df = 1, p-value = 3.391e-09
```

```r
p.author.bar <- t.author.bar / sum(t.author.bar)
dimnames(p.author.bar) <- NULL
ES.h(p.author.bar[1], p.author.bar[2])
```

```
## [1] 1.197563
```

```r
t.author.line <- table(subset(line, treatment == "Treatment" & Strength == "High")$author_aligned)
chisq.test(t.author.line)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  t.author.line
## X-squared = 55.309, df = 1, p-value = 1.03e-13
```

```r
p.author.line <- t.author.line / sum(t.author.line)
dimnames(p.author.line) <- NULL
ES.h(p.author.line[1], p.author.line[2])
```

```
## [1] 1.576416
```

```r
df %>% group_by(ChartType, author_aligned) %>%
  summarise(n = n())
```

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

```
## # A tibble: 6 x 3
## # Groups:   ChartType [2]
##   ChartType author_aligned     n
##   <chr>     <chr>          <int>
## 1 Bar       Aligned          154
## 2 Bar       No-Side          110
## 3 Bar       Unaligned         66
## 4 Line      Aligned          157
## 5 Line      No-Side          110
## 6 Line      Unaligned         63
```

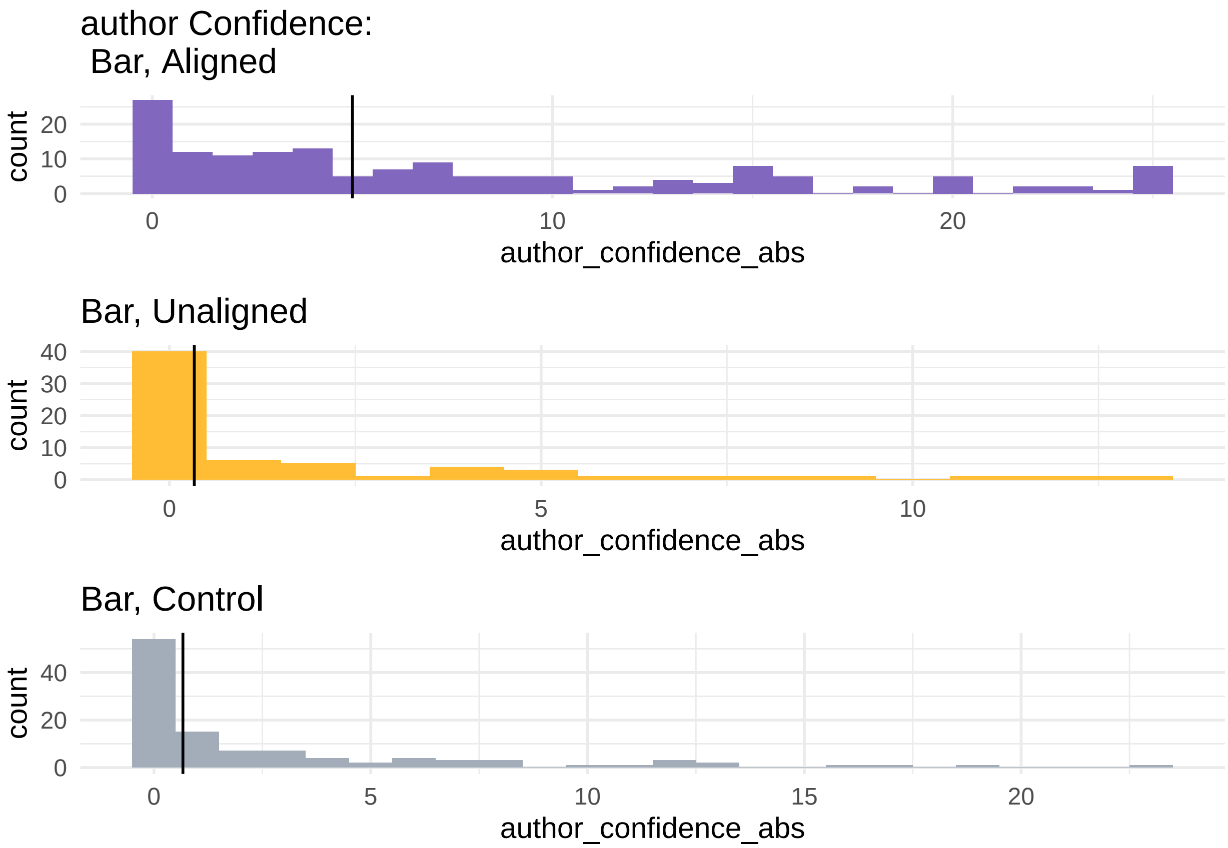
```r
df %>% group_by(Strength, author_aligned) %>%
  summarise(n = n())
```

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

```
## # A tibble: 5 x 3
## # Groups:   Strength [3]
##   Strength author_aligned     n
##   <chr>    <chr>          <int>
## 1 High     Aligned          180
## 2 High     Unaligned         40
## 3 Low      Aligned          131
## 4 Low      Unaligned         89
## 5 No-Side  No-Side          220
```

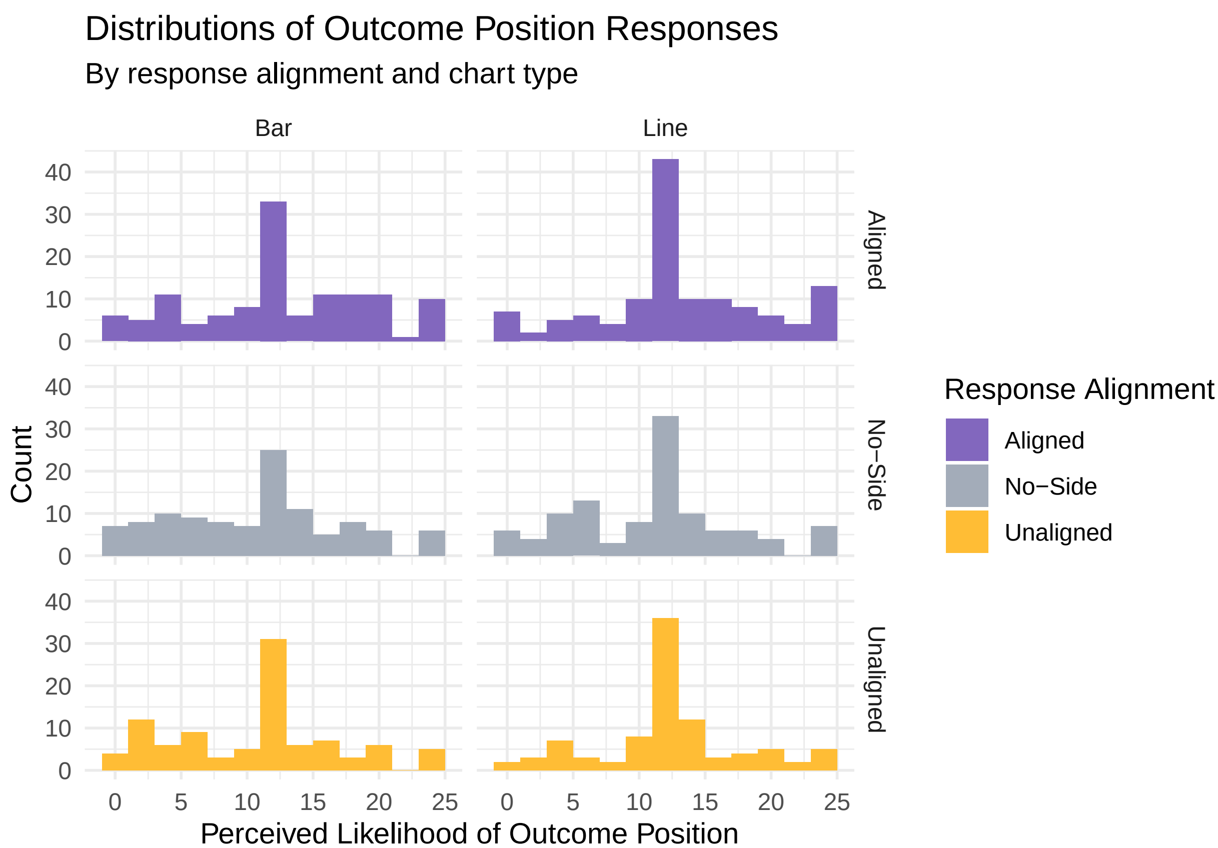
```r
# - Bar Chart, author Aligned, Treatment Condition
bar_aligned_treatment_author = subset(bar, author_aligned == "Aligned" & treatment == "Treatment")
bar_aligned_treatment_author_plot =
  ggplot(bar_aligned_treatment_author, aes(x = author_confidence_abs, y = after_stat(count)))+
  geom_histogram(binwidth = 1, fill = a)+
  labs(
    #title = "author Confidence: Bar Chart, Treatment Condition, Aligned with Slant"
    title = "author Confidence:\n Bar, Aligned"
  )

# - Bar Chart, author Unaligned, Treatment Condition
bar_unaligned_treatment_author = subset(bar, author_aligned == "Unaligned" & treatment == "Treatment")
```

```r
bar_unaligned_treatment_author_plot =
  ggplot(bar_unaligned_treatment_author, aes(x = author_confidence_abs, y = after_stat(count)))+
  geom_histogram(binwidth = 1, fill = u)+
  labs(
    #title = "author Confidence: Bar Chart, Treatment Condition, Unaligned with Slant"
    title = "Bar, Unaligned"
  )

# - Bar Chart, Control Condition
bar_control = subset(bar, treatment == "Control")
bar_control_author_plot =
  ggplot(bar_control, aes(x = author_confidence_abs, y = after_stat(count)))+
  geom_histogram(binwidth = 1, fill = c)+
  labs(
    #title = "author Confidence: Bar Chart, Control Condition"
    title = "Bar, Control"
  )

# - Line Chart, author Aligned, Treatment Condition
line_aligned_treatment_author = subset(line, author_aligned == "Aligned" & treatment == "Treatment")
line_aligned_treatment_author_plot =
  ggplot(line_aligned_treatment_author, aes(x = author_confidence_abs, y = after_stat(count)))+
  geom_histogram(binwidth = 1, fill = a)+
  labs(
    #title = "author Confidence: Line Chart, Treatment Condition, Aligned with Slant"
    title = "author Confidence:\n Line, Aligned"
  )

# - Line Chart, author Unaligned, Treatment Condition
line_unaligned_treatment_author = subset(line, author_aligned == "Unaligned" & treatment == "Treatment")
line_unaligned_treatment_author_plot =
  ggplot(line_unaligned_treatment_author, aes(x = author_confidence_abs, y = after_stat(count)))+
  geom_histogram(binwidth = 1, fill = u)+
  labs(
    #title = "author Confidence: Line Chart, Treatment Condition, Unaligned with Slant"
    title = "Line, Unaligned"
  )

# - Line Chart, Control Condition
line_control = subset(line, treatment == "Control")
line_control_author_plot =
  ggplot(line_control, aes(x = author_confidence_abs, y = after_stat(count)))+
  geom_histogram(binwidth = 1, fill = c)+
  labs(
    #title = "author Confidence: Line Chart, Control Condition"
    title = "Line, Control"
  )


ggarrange(bar_aligned_treatment_author_plot + geom_vline(xintercept = median(bar_aligned_treatment_autho
          bar_unaligned_treatment_author_plot + geom_vline(xintercept = median(bar_unaligned_treatment_a
          bar_control_author_plot + geom_vline(xintercept = median(bar_control$author_confidence_abs)),
          ncol = 1)
```




```
ggarrange(line_aligned_treatment_author_plot + geom_vline(xintercept = median(line_aligned_treatment_au
          line_unaligned_treatment_author_plot + geom_vline(xintercept = median(line_unaligned_treatmen
          line_control_author_plot + geom_vline(xintercept = median(line_control$author_confidence_abs)
          ncol = 1)
```


```r
ggplot(df, aes(x = outcome_confidence_abs, y = after_stat(count)))+
  geom_histogram(binwidth = 2, aes(fill = outcome_aligned))+
  labs(
    title = "Distributions of Outcome Position Responses",
    subtitle = "By response alignment and chart type",
    x = "Perceived Likelihood of Outcome Position",
    y = "Count",
    fill = "Response Alignment"
  )+
  scale_fill_manual(values = c(a, c, u))+
  facet_grid(outcome_aligned ~ ChartType)
```




```
df %>%
  group_by(ChartType, outcome_aligned) %>%
  summarise(
    mean = mean(outcome_confidence_abs)
  )
```

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

```
## # A tibble: 6 x 3
## # Groups:   ChartType [2]
##   ChartType outcome_aligned  mean
##   <chr>     <chr>           <dbl>
## 1 Bar       Aligned          13.1
## 2 Bar       No-Side          11.3
## 3 Bar       Unaligned        11.3
## 4 Line      Aligned          13.7
## 5 Line      No-Side          11.7
## 6 Line      Unaligned        12.9
```

```
kruskal.test(outcome_confidence_abs ~ as.factor(outcome_aligned), data = df)
```

```
##
##  Kruskal-Wallis rank sum test
```

```
##
## data:  outcome_confidence_abs by as.factor(outcome_aligned)
## Kruskal-Wallis chi-squared = 11.234, df = 2, p-value = 0.003636
```

```r
kruskal_effsize(outcome_confidence_abs ~ as.factor(outcome_aligned), data = df)
```

```
## # A tibble: 1 x 5
##   .y.                        n effsize method  magnitude
## * <chr>                  <int>   <dbl> <chr>   <ord>
## 1 outcome_confidence_abs   660  0.0141 eta2[H] small
```

```r
dunnTest(outcome_confidence_abs ~ as.factor(outcome_aligned), data = df, method = "bonferroni")
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
##   p-values adjusted with the Bonferroni method.
```

```
##            Comparison          Z     P.unadj       P.adj
## 1   Aligned - No-Side  3.2542069 0.001137094 0.003411282
## 2 Aligned - Unaligned  2.1749509 0.029633793 0.088901380
## 3 No-Side - Unaligned -0.9183423 0.358439674 1.000000000
```

```r
kruskal.test(outcome_confidence_abs ~ as.factor(outcome_aligned), data = bar)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  outcome_confidence_abs by as.factor(outcome_aligned)
## Kruskal-Wallis chi-squared = 6.2552, df = 2, p-value = 0.04382
```

```r
kruskal_effsize(outcome_confidence_abs ~ as.factor(outcome_aligned), data = bar)
```

```
## # A tibble: 1 x 5
##   .y.                        n effsize method  magnitude
## * <chr>                  <int>   <dbl> <chr>   <ord>
## 1 outcome_confidence_abs   330  0.0130 eta2[H] small
```

```r
dunnTest(outcome_confidence_abs ~ as.factor(outcome_aligned), data = bar, method = "bonferroni")
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
##   p-values adjusted with the Bonferroni method.
```

```
##            Comparison          Z    P.unadj      P.adj
## 1   Aligned - No-Side 2.13006535 0.03316622 0.09949866
## 2 Aligned - Unaligned 2.13819133 0.03250122 0.09750365
## 3 No-Side - Unaligned 0.07770116 0.93806577 1.00000000
```

```r
kruskal.test(outcome_confidence_abs ~ as.factor(outcome_aligned), data = line)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  outcome_confidence_abs by as.factor(outcome_aligned)
## Kruskal-Wallis chi-squared = 6.2759, df = 2, p-value = 0.04337
```

```
kruskal_effsize(outcome_confidence_abs ~ as.factor(outcome_aligned), data = line)
```

```
## # A tibble: 1 x 5
##   .y.                        n effsize method  magnitude
## * <chr>                  <int>   <dbl> <chr>   <ord>
## 1 outcome_confidence_abs   330  0.0131 eta2[H] small
```

```
dunnTest(outcome_confidence_abs ~ as.factor(outcome_aligned), data = line, method = "bonferroni")
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
##   p-values adjusted with the Bonferroni method.
```

```
##            Comparison          Z    P.unadj      P.adj
## 1   Aligned - No-Side  2.4875959 0.01286098 0.03858294
## 2 Aligned - Unaligned  0.8275224 0.40794103 1.00000000
## 3 No-Side - Unaligned -1.4886047 0.13659149 0.40977448
```

```
ggplot(df, aes(x = author_confidence_abs, y = after_stat(count)))+
  geom_histogram(binwidth = 2, aes(fill = author_aligned))+
  labs(
    title = "Distributions of Author Position Responses",
    subtitle = "By response alignment and chart type",
    x = "Perceived Likelihood of Author Position",
    y = "Count",
    fill = "Response Alignment"
  )+
  scale_fill_manual(values = c(a, c, u))+
  facet_grid(author_aligned ~ ChartType)
```


```r
df %>%
  group_by(ChartType, author_aligned) %>%
  summarise(
    mean = mean(author_confidence_abs)
  )
```

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

```
## # A tibble: 6 x 3
## # Groups:   ChartType [2]
##   ChartType author_aligned  mean
##   <chr>     <chr>          <dbl>
## 1 Bar       Aligned         7.54
## 2 Bar       No-Side         2.78
## 3 Bar       Unaligned       1.77
## 4 Line      Aligned         7.51
## 5 Line      No-Side         2.99
## 6 Line      Unaligned       3.03
```

```r
kruskal.test(author_confidence_abs ~ as.factor(author_aligned), data = bar)
```

```
##
##  Kruskal-Wallis rank sum test
```

```
##
## data:  author_confidence_abs by as.factor(author_aligned)
## Kruskal-Wallis chi-squared = 63.382, df = 2, p-value = 1.725e-14
```

```
kruskal_effsize(author_confidence_abs ~ as.factor(author_aligned), data = bar)
```

```
## # A tibble: 1 x 5
##   .y.                       n effsize method  magnitude
## * <chr>                 <int>   <dbl> <chr>   <ord>
## 1 author_confidence_abs   330   0.188 eta2[H] large
```

```
dunnTest(author_confidence_abs ~ as.factor(author_aligned), data = bar, method = "bonferroni")
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
##   p-values adjusted with the Bonferroni method.

##            Comparison        Z      P.unadj        P.adj
## 1   Aligned - No-Side 6.376884 1.807272e-10 5.421817e-10
## 2 Aligned - Unaligned 6.713019 1.906387e-11 5.719160e-11
## 3 No-Side - Unaligned 1.230325 2.185756e-01 6.557267e-01
```

```
kruskal.test(author_confidence_abs ~ as.factor(author_aligned), data = line)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  author_confidence_abs by as.factor(author_aligned)
## Kruskal-Wallis chi-squared = 51.246, df = 2, p-value = 7.447e-12
```

```
kruskal_effsize(author_confidence_abs ~ as.factor(author_aligned), data = line)
```

```
## # A tibble: 1 x 5
##   .y.                       n effsize method  magnitude
## * <chr>                 <int>   <dbl> <chr>   <ord>
## 1 author_confidence_abs   330   0.151 eta2[H] large
```

```
dunnTest(author_confidence_abs ~ as.factor(author_aligned), data = line, method = "bonferroni")
```

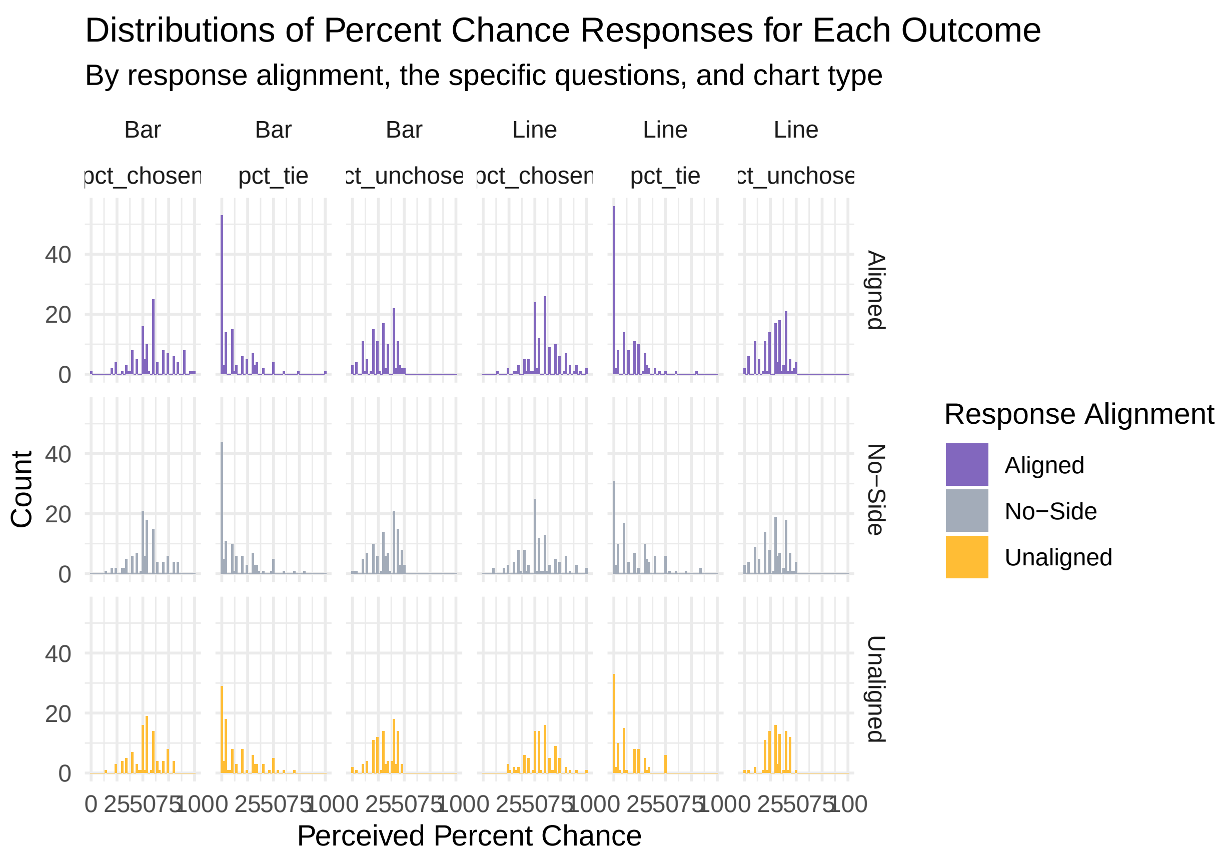
```
## Dunn (1964) Kruskal-Wallis multiple comparison
##   p-values adjusted with the Bonferroni method.

##            Comparison         Z      P.unadj        P.adj
## 1   Aligned - No-Side 6.1813901 6.353958e-10 1.906187e-09
## 2 Aligned - Unaligned 5.5141637 3.504423e-08 1.051327e-07
## 3 No-Side - Unaligned 0.3404139 7.335449e-01 1.000000e+00
```

These have been replications of the hypotheses from Study 1. We have additional hypotheses regarding the percent chances, which were not collected in Study 1. With these, we evaluate this effect using another type of outcome response.

Hypothesis 1.2 (new variables)

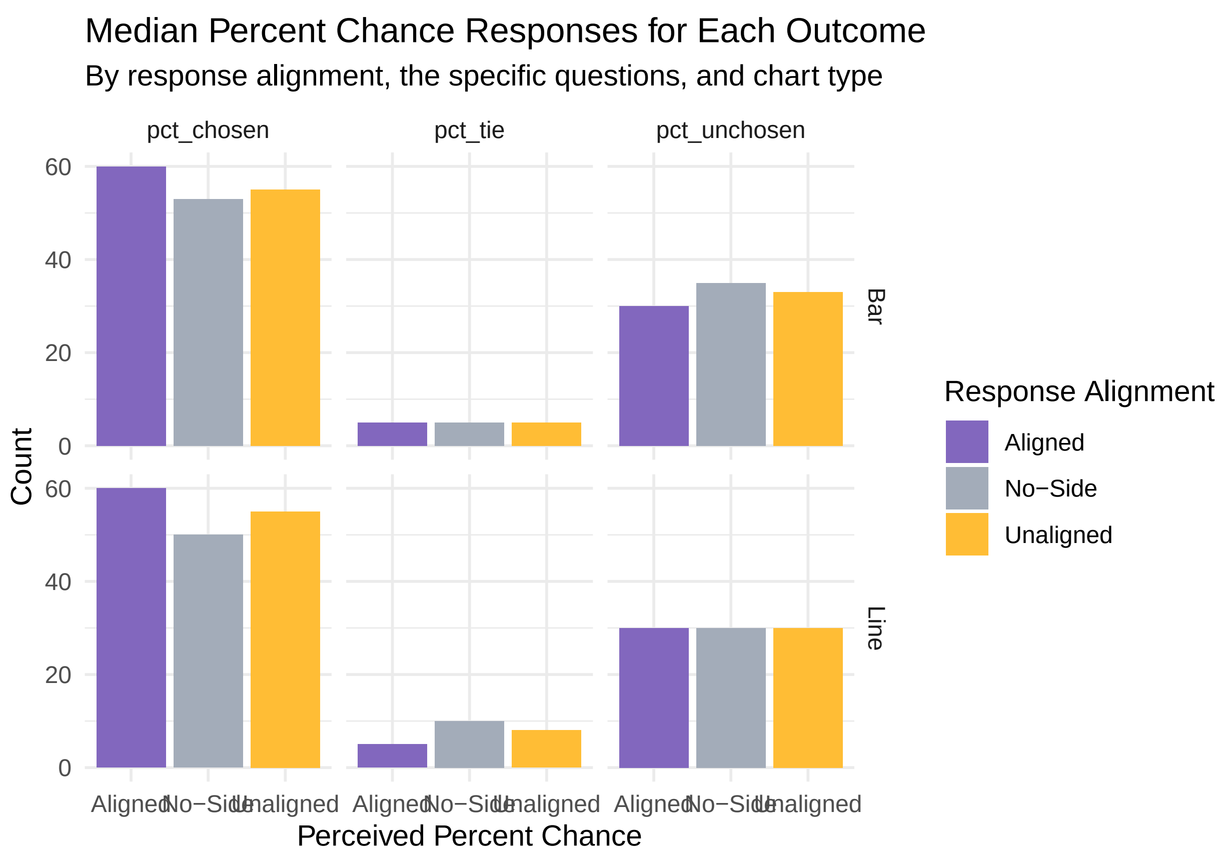
1.3. Participants who respond in alignment with the slant presented estimate a higher likelihood for their selected outcome than participants who are unaligned. 1.4. Participants who respond in alignment with the slant presented estimate a lower likelihood for the non-selected outcome than participants who are unaligned. 1.5. Participants who respond in alignment with the slant presented estimate a lower likelihood for a tie than participants who are unaligned.

```
ggplot(df %>% gather(key = pct_source, value = pct_response, pct_chosen, pct_unchosen, pct_tie),
       aes(x = pct_response, y = after_stat(count)))+
  geom_histogram(binwidth = 2, aes(fill = outcome_aligned))+
  labs(
    title = "Distributions of Percent Chance Responses for Each Outcome",
    subtitle = "By response alignment, the specific questions, and chart type",
    x = "Perceived Percent Chance",
    y = "Count",
    fill = "Response Alignment"
  )+
  scale_fill_manual(values = c(a, c, u))+
  facet_grid(outcome_aligned ~ ChartType + pct_source)
```




```
ggplot(df %>% gather(key = pct_source, value = pct_response, pct_chosen, pct_unchosen, pct_tie),
       aes(y = pct_response, x = outcome_aligned, fill = outcome_aligned))+
  geom_bar(stat = "summary", fun = "median")+
  labs(
```

```r
    title = "Median Percent Chance Responses for Each Outcome",
    subtitle = "By response alignment, the specific questions, and chart type",
    x = "Perceived Percent Chance",
    y = "Count",
    fill = "Response Alignment"
  )+
  scale_fill_manual(values = c(a, c, u))+
  facet_grid(ChartType ~ pct_source)
```




```r
wilcox.test(pct_chosen ~ as.factor(outcome_aligned), data = subset(bar, treatment == "Treatment"))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pct_chosen by as.factor(outcome_aligned)
## W = 6982.5, p-value = 0.02925
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(pct_chosen ~ as.factor(outcome_aligned), data = subset(line, treatment == "Treatment"))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pct_chosen by as.factor(outcome_aligned)
```

```
## W = 6521.5, p-value = 0.1711
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(pct_unchosen ~ as.factor(outcome_aligned), data = subset(bar, treatment == "Treatment"))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pct_unchosen by as.factor(outcome_aligned)
## W = 5294.5, p-value = 0.15
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(pct_unchosen ~ as.factor(outcome_aligned), data = subset(line, treatment == "Treatment"))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pct_unchosen by as.factor(outcome_aligned)
## W = 5285, p-value = 0.1928
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(pct_tie ~ as.factor(outcome_aligned), data = subset(bar, treatment == "Treatment"))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pct_tie by as.factor(outcome_aligned)
## W = 5261.5, p-value = 0.1245
## alternative hypothesis: true location shift is not equal to 0
```

```
wilcox.test(pct_tie ~ as.factor(outcome_aligned), data = subset(line, treatment == "Treatment"))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  pct_tie by as.factor(outcome_aligned)
## W = 5625, p-value = 0.5626
## alternative hypothesis: true location shift is not equal to 0
```

After investigating these initial hypotheses, we move on to the hypotheses examining the role of the bias condition.

Hypotheses 4.1-4.2

4.1. Participants who viewed stronger language and were aligned with the chart outcome presented were more confident in their outcome responses than those who viewed weaker language (as well as control).

4.2. Participants who viewed stronger language and were unaligned with the chart outcome presented were more confident in their outcome responses than those who viewed weaker language (no different than control).

```r
kruskal.test(author_confidence_abs ~ as.factor(outcome_aligned), data = subset(bar, Strength = "High"))
```

```
## Warning: In subset.data.frame(bar, Strength = "High") :
##  extra argument 'Strength' will be disregarded

## Warning: In subset.data.frame(bar, Strength = "High") :
##  extra argument 'Strength' will be disregarded

##
##  Kruskal-Wallis rank sum test
##
## data:  author_confidence_abs by as.factor(outcome_aligned)
## Kruskal-Wallis chi-squared = 19.826, df = 2, p-value = 4.952e-05
```

```r
kruskal.test(author_confidence_abs ~ as.factor(outcome_aligned), data = subset(bar, Strength = "Low"))
```

```
## Warning: In subset.data.frame(bar, Strength = "Low") :
##  extra argument 'Strength' will be disregarded

## Warning: In subset.data.frame(bar, Strength = "Low") :
##  extra argument 'Strength' will be disregarded

##
##  Kruskal-Wallis rank sum test
##
## data:  author_confidence_abs by as.factor(outcome_aligned)
## Kruskal-Wallis chi-squared = 19.826, df = 2, p-value = 4.952e-05
```

```r
kruskal.test(author_confidence_abs ~ as.factor(outcome_aligned), data = subset(line, Strength = "High")
```

```
## Warning: In subset.data.frame(line, Strength = "High") :
##  extra argument 'Strength' will be disregarded

## Warning: In subset.data.frame(line, Strength = "High") :
##  extra argument 'Strength' will be disregarded

##
##  Kruskal-Wallis rank sum test
##
## data:  author_confidence_abs by as.factor(outcome_aligned)
## Kruskal-Wallis chi-squared = 24.027, df = 2, p-value = 6.06e-06
```

```r
kruskal.test(author_confidence_abs ~ as.factor(outcome_aligned), data = subset(line, Strength = "Low"))
```

```
## Warning: In subset.data.frame(line, Strength = "Low") :
##  extra argument 'Strength' will be disregarded

## Warning: In subset.data.frame(line, Strength = "Low") :
##  extra argument 'Strength' will be disregarded
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  author_confidence_abs by as.factor(outcome_aligned)
## Kruskal-Wallis chi-squared = 24.027, df = 2, p-value = 6.06e-06
```

```
# aligned, bar
kruskal.test(outcome_confidence_abs ~ as.factor(Strength), data = subset(bar, outcome_aligned != "Unali
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  outcome_confidence_abs by as.factor(Strength)
## Kruskal-Wallis chi-squared = 5.1099, df = 2, p-value = 0.0777
```

```
kruskal_effsize(outcome_confidence_abs ~ as.factor(Strength), data = subset(bar, outcome_aligned != "Una
```

```
## # A tibble: 1 x 5
##   .y.                        n effsize method  magnitude
## * <chr>                  <int>   <dbl> <chr>   <ord>
## 1 outcome_confidence_abs   233  0.0135 eta2[H] small
```

```
dunnTest(outcome_confidence_abs ~ as.factor(Strength), data = subset(bar, outcome_aligned != "Unaligned
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Bonferroni method.

##       Comparison          Z    P.unadj      P.adj
## 1     High - Low -0.8369132 0.40264134 1.00000000
## 2 High - No-Side  1.1605407 0.24582873 0.73748618
## 3  Low - No-Side  2.2262821 0.02599529 0.07798587
```

```
# aligned, line
kruskal.test(outcome_confidence_abs ~ as.factor(Strength), data = subset(line, outcome_aligned != "Unal
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  outcome_confidence_abs by as.factor(Strength)
## Kruskal-Wallis chi-squared = 6.5306, df = 2, p-value = 0.03819
```

```
kruskal_effsize(outcome_confidence_abs ~ as.factor(Strength), data = subset(line, outcome_aligned != "Ur
```

```
## # A tibble: 1 x 5
##   .y.                        n effsize method  magnitude
## * <chr>                  <int>   <dbl> <chr>   <ord>
## 1 outcome_confidence_abs   238  0.0193 eta2[H] small
```

```r
dunnTest(outcome_confidence_abs ~ as.factor(Strength), data = subset(line, outcome_aligned != "Unaligne
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
##   p-values adjusted with the Bonferroni method.

##       Comparison         Z    P.unadj     P.adj
## 1     High - Low -0.737173 0.46101715 1.0000000
## 2 High - No-Side  1.679192 0.09311467 0.2793440
## 3  Low - No-Side  2.408063 0.01603740 0.0481122
```

```r
# unaligned, bar
kruskal.test(outcome_confidence_abs ~ as.factor(Strength), data = subset(bar, outcome_aligned != "Align
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  outcome_confidence_abs by as.factor(Strength)
## Kruskal-Wallis chi-squared = 0.64165, df = 2, p-value = 0.7256
```

```r
kruskal_effsize(outcome_confidence_abs ~ as.factor(Strength), data = subset(bar, outcome_aligned != "Al
```

```
## # A tibble: 1 x 5
##   .y.                        n  effsize method  magnitude
## * <chr>                  <int>    <dbl> <chr>   <ord>
## 1 outcome_confidence_abs   207 -0.00666 eta2[H] small
```

```r
dunnTest(outcome_confidence_abs ~ as.factor(Strength), data = subset(bar, outcome_aligned != "Aligned")
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
##   p-values adjusted with the Bonferroni method.

##       Comparison          Z   P.unadj P.adj
## 1     High - Low  0.7998003 0.4238265     1
## 2 High - No-Side  0.3922841 0.6948483     1
## 3  Low - No-Side -0.5464065 0.5847866     1
```

```r
# unaligned, line
kruskal.test(outcome_confidence_abs ~ as.factor(Strength), data = subset(line, outcome_aligned != "Alig
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  outcome_confidence_abs by as.factor(Strength)
## Kruskal-Wallis chi-squared = 2.4221, df = 2, p-value = 0.2979
```

```r
kruskal_effsize(outcome_confidence_abs ~ as.factor(Strength), data = subset(line, outcome_aligned != "Al
```

```
## # A tibble: 1 x 5
##   .y.                        n effsize method  magnitude
## * <chr>                  <int>   <dbl> <chr>   <ord>
## 1 outcome_confidence_abs   202 0.00212 eta2[H] small
```

```r
dunnTest(outcome_confidence_abs ~ as.factor(Strength), data = subset(line, outcome_aligned != "Aligned")
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
##   p-values adjusted with the Bonferroni method.

##        Comparison          Z   P.unadj     P.adj
## 1     High - Low -0.2741915 0.7839374 1.0000000
## 2 High - No-Side  1.0086124 0.3131606 0.9394817
## 3  Low - No-Side  1.4289139 0.1530290 0.4590870
```

```r
df %>% group_by(ChartType, Strength, outcome_aligned) %>%
  summarise(
    mean = mean(outcome_confidence_abs)
  )
```

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

## # A tibble: 10 x 4
## # Groups:   ChartType, Strength [6]
##    ChartType Strength outcome_aligned  mean
##    <chr>     <chr>    <chr>           <dbl>
##  1 Bar       High     Aligned          12.6
##  2 Bar       High     Unaligned        11.8
##  3 Bar       Low      Aligned          13.5
##  4 Bar       Low      Unaligned        10.6
##  5 Bar       No-Side  No-Side          11.3
##  6 Line      High     Aligned          13.4
##  7 Line      High     Unaligned        12.4
##  8 Line      Low      Aligned          14.1
##  9 Line      Low      Unaligned        13.3
## 10 Line      No-Side  No-Side          11.7
```

```r
subset(df, treatment == "Treatment") %>% group_by(ChartType, Strength, outcome_aligned) %>%
  summarise(
    n = n()
  )
```

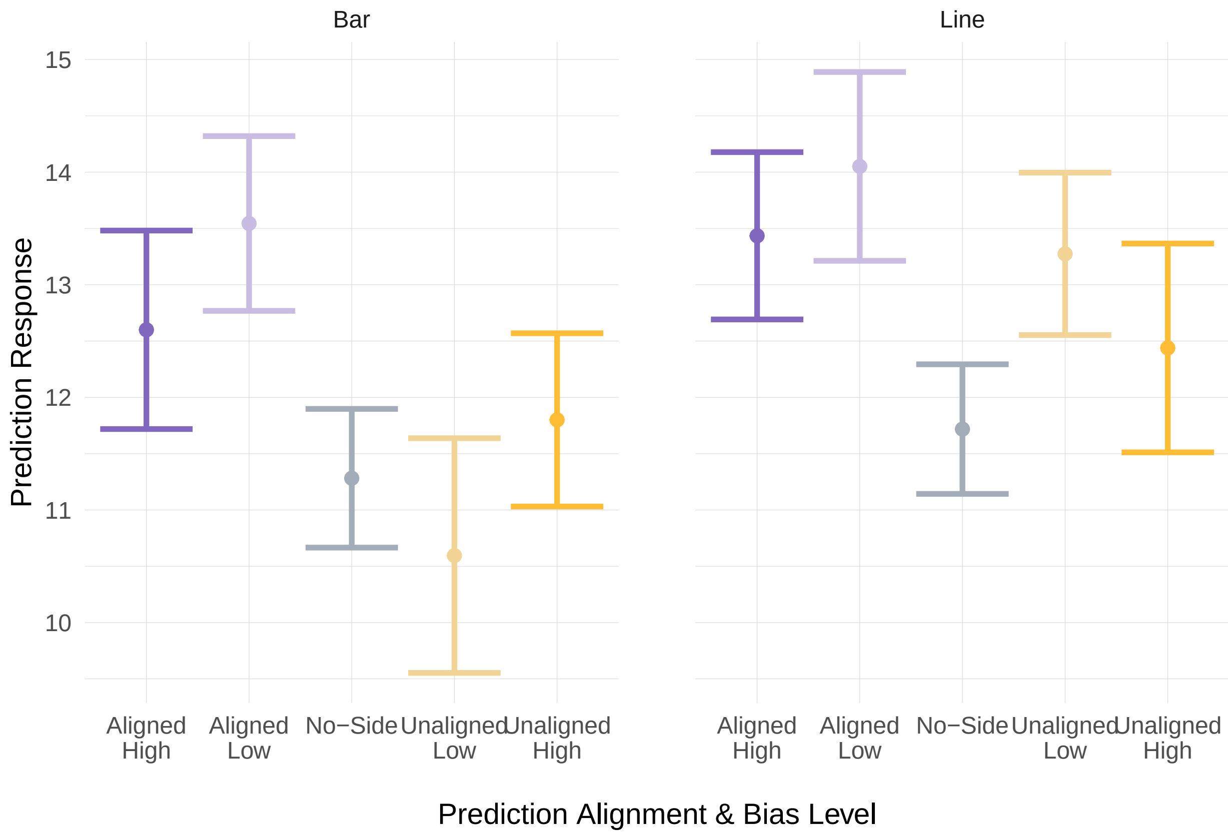
```
## `summarise()` has grouped output by 'ChartType', 'Strength'. You can override
## using the `.groups` argument.

## # A tibble: 8 x 4
## # Groups:   ChartType, Strength [4]
##   ChartType Strength outcome_aligned     n
##   <chr>     <chr>    <chr>           <int>
## 1 Bar       High     Aligned            55
## 2 Bar       High     Unaligned          55
## 3 Bar       Low      Aligned            68
## 4 Bar       Low      Unaligned          42
## 5 Line      High     Aligned            69
## 6 Line      High     Unaligned          41
## 7 Line      Low      Aligned            59
## 8 Line      Low      Unaligned          51
```

```r
condition_vis_outcome <- df %>%
  group_by(ChartType, outcome_aligned, Strength) %>%
  summarize(
    mean_outcome = mean(outcome_confidence_abs),
    sd_outcome = sd(outcome_confidence_abs),
    n_outcome = n(),
    se_outcome = sd_outcome / sqrt(n_outcome)
  ) %>%
  mutate(
    condition = paste(outcome_aligned, Strength)
  )
```

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

```r
ggplot(condition_vis_outcome, aes(x = condition, y = mean_outcome, color = condition))+
  geom_point(size = 2)+
  geom_errorbar(aes(ymin = mean_outcome - se_outcome, ymax = mean_outcome + se_outcome), size = 1)+
  scale_x_discrete(limits = c("Aligned High", "Aligned Low", "No-Side No-Side", "Unaligned Low", "Unali
  scale_color_manual(values = c(a, a1, c, u, u1))+
    labs(color = "Prediction Alignment &\nBias Level",
       y = "Prediction Response", x = "\nPrediction Alignment & Bias Level")+
  facet_grid(.~ChartType )+
  theme(legend.position = 'none',
        panel.spacing = unit(2, "lines"),
  panel.grid.major = element_line(color = "grey85", size = 0.1),  # Lighten and thin major gridlines
  panel.grid.minor = element_line(color = "grey85", size = 0.05))  # Lighten and thin minor gridlines
```

```
## 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.
```

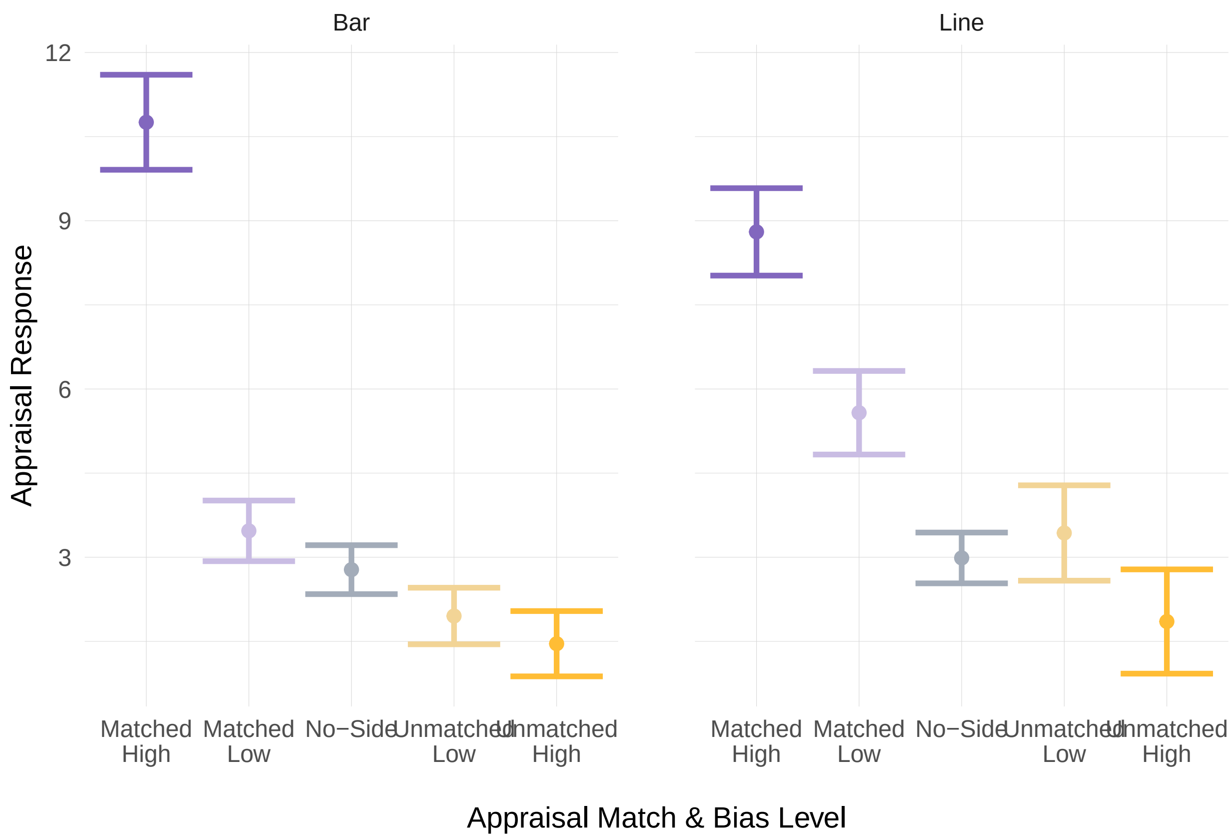



```r
condition_vis_author <- df %>%
  group_by(ChartType, author_aligned, Strength) %>%
  summarize(
    mean_author = mean(author_confidence_abs),
    sd_author = sd(author_confidence_abs),
    n_author = n(),
    se_author = sd_author / sqrt(n_author)
  ) %>%
  mutate(
    condition = paste(author_aligned, Strength)
  )
```

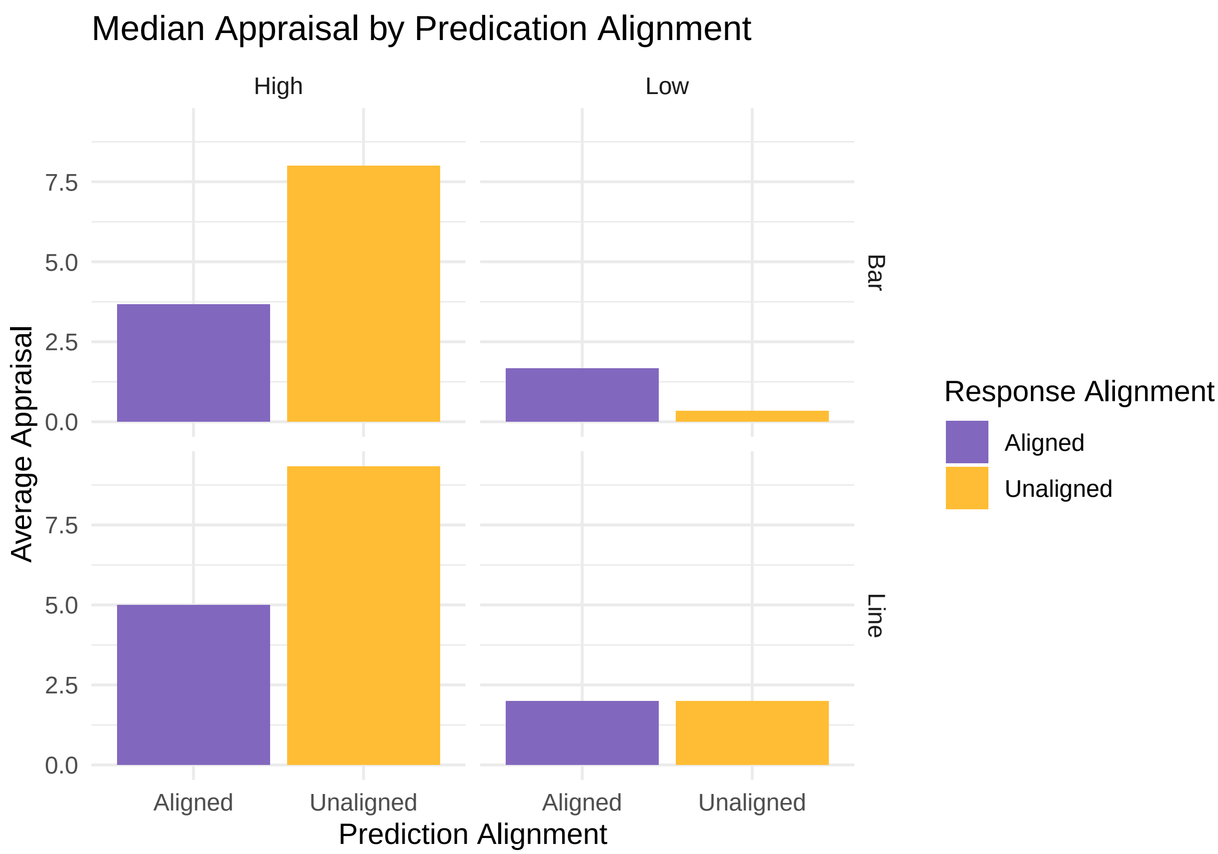
```
## `summarise()` has grouped output by 'ChartType', 'author_aligned'. You can
## override using the `.groups` argument.
```

```r
ggplot(condition_vis_author, aes(x = condition, y = mean_author, color = condition))+
  geom_point(size = 2)+
  geom_errorbar(aes(ymin = mean_author - se_author, ymax = mean_author + se_author), size =1)+
  scale_x_discrete(limits = c("Aligned High", "Aligned Low", "No-Side No-Side", "Unaligned Low", "Unali
  scale_color_manual(values = c(a, a1, c, u, u1), labels = c("Matched High", "Matched Low", "No-Side",
  labs(color = "Appraisal Match &\nBias Level",
       y = "Appraisal Response", x = "\nAppraisal Match & Bias Level")+
  facet_grid(.~ChartType )+
  theme(legend.position = 'none',
        panel.spacing = unit(2, "lines"),
```

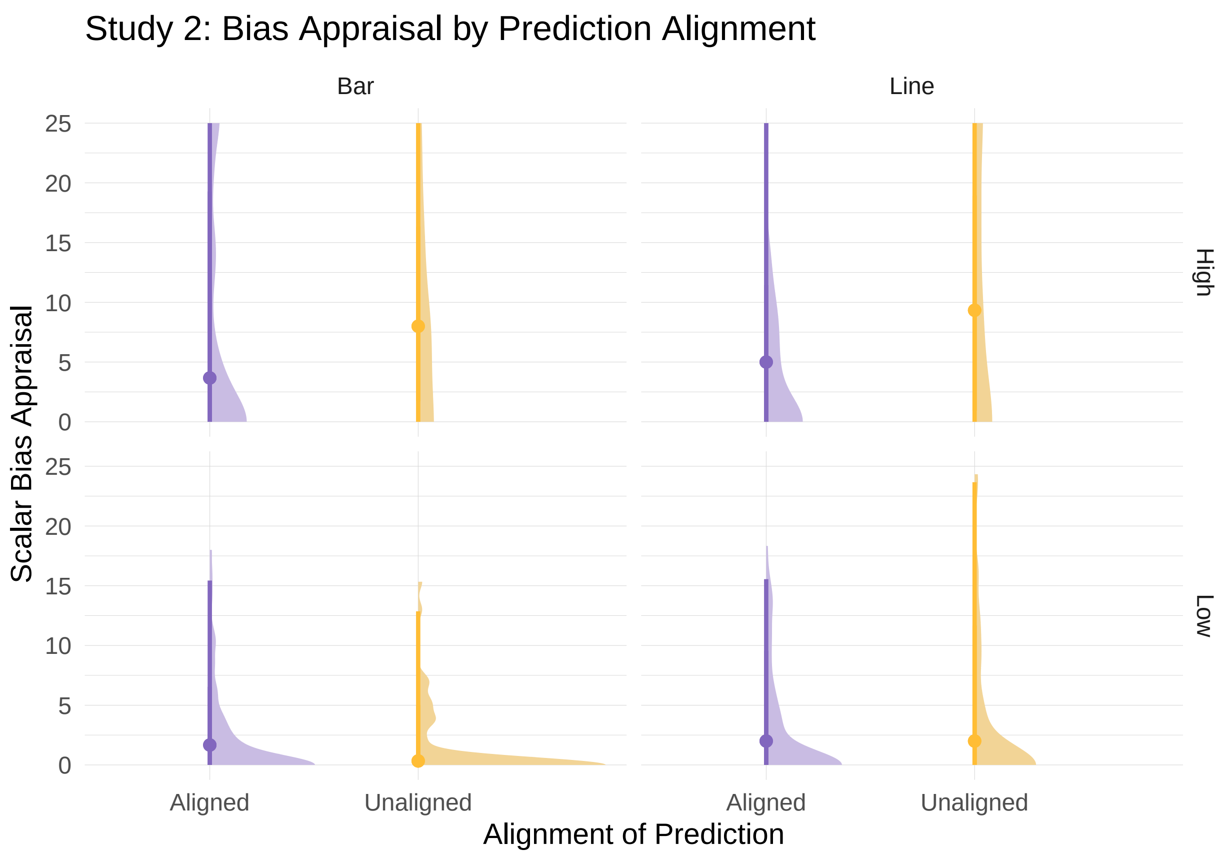
```r
  panel.grid.major = element_line(color = "grey85", size = 0.1),  # Lighten and thin major gridlines
  panel.grid.minor = element_line(color = "grey85", size = 0.05))  # Lighten and thin minor gridlines
```




```r
ggplot(subset(df, treatment == "Treatment"), aes(x = outcome_aligned, y = author_confidence_abs, fill =
  geom_bar(stat = "summary", fun = "median")+
  scale_fill_manual(values = c(a, u))+
  labs(
    title = "Median Appraisal by Predication Alignment",
    # subtitle = "Did people perceive the author as having a more extreme position if they disagreed wi
    y = "Average Appraisal",
    x = "Prediction Alignment",
    fill = "Response Alignment"
  )+
  facet_grid(ChartType ~ Strength)
```




```
ggplot(subset(df, treatment == "Treatment"), aes(x = outcome_aligned, y = author_confidence_abs, fill =
  stat_halfeye(size = 2)+
  scale_fill_manual(values = c(a1, u1))+
  scale_color_manual(values = c(a, u))+
  labs(
    title = "Study 2: Bias Appraisal by Prediction Alignment",
    # subtitle = "Did people perceive the author as having a more extreme position\nif they disagreed w
    y = "Scalar Bias Appraisal",
    x = "Alignment of Prediction"
  )+
  facet_grid(Strength ~ChartType)+
  theme(legend.position = 'none',
  panel.grid.major = element_line(color = "grey85", size = 0.1),  # Lighten and thin major gridlines
  panel.grid.minor = element_line(color = "grey85", size = 0.05))  # Lighten and thin minor gridlines
```




```
kruskal.test(author_confidence_abs ~ outcome_aligned, data = subset(bar, Strength == "High"))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  author_confidence_abs by outcome_aligned
## Kruskal-Wallis chi-squared = 2.9127, df = 1, p-value = 0.08789
```

```
kruskal.test(author_confidence_abs ~ outcome_aligned, data = subset(bar, Strength == "Low"))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  author_confidence_abs by outcome_aligned
## Kruskal-Wallis chi-squared = 2.0933, df = 1, p-value = 0.1479
```

```
kruskal.test(author_confidence_abs ~ outcome_aligned, data = subset(line, Strength == "High"))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  author_confidence_abs by outcome_aligned
## Kruskal-Wallis chi-squared = 5.9486, df = 1, p-value = 0.01473
```

```r
kruskal.test(author_confidence_abs ~ outcome_aligned, data = subset(line, Strength == "Low"))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  author_confidence_abs by outcome_aligned
## Kruskal-Wallis chi-squared = 0.85217, df = 1, p-value = 0.3559
```

Stepwise model

Comparing the priming effect of the outcome/bias question order

```r
#### Prediction ####

model0_outcome <- lmer(outcome_confidence_abs ~ ChartType + (1|Order) + (1|Slant), data = df)

model1_outcome <- lmer(outcome_confidence_abs ~ ChartType + outcome_aligned + (1|Order) + (1|Slant), da

model2_outcome <- lmer(outcome_confidence_abs ~ ChartType + outcome_aligned + Strength + (1|Order) + (1
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

```r
model3_outcome <- lmer(outcome_confidence_abs ~ ChartType + outcome_aligned + Strength + Prime + (1|Ord
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

```r
model4_outcome <- lmer(outcome_confidence_abs ~ ChartType + outcome_aligned * Strength + Prime + (1|Ord
```

```
## fixed-effect model matrix is rank deficient so dropping 4 columns / coefficients
```

```r
model5_outcome <- lmer(outcome_confidence_abs ~ ChartType + outcome_aligned * Strength * Prime + (1|Ord
```

```
## fixed-effect model matrix is rank deficient so dropping 8 columns / coefficients

## boundary (singular) fit: see help('isSingular')
```

```r
model6_outcome <- lmer(outcome_confidence_abs ~ ChartType * outcome_aligned * Strength * Prime + (1|Ord
```

```
## fixed-effect model matrix is rank deficient so dropping 16 columns / coefficients
```

```r
model7_outcome <- lmer(outcome_confidence_abs ~ ChartType * outcome_aligned * Strength * Prime + age +
```

```
## fixed-effect model matrix is rank deficient so dropping 16 columns / coefficients
## boundary (singular) fit: see help('isSingular')
```

```r
anova(model0_outcome, model1_outcome, model2_outcome, model3_outcome, model4_outcome, model5_outcome, mo
```

```
## refitting model(s) with ML (instead of REML)

## Data: df
## Models:
## model0_outcome: outcome_confidence_abs ~ ChartType + (1 | Order) + (1 | Slant)
## model1_outcome: outcome_confidence_abs ~ ChartType + outcome_aligned + (1 | Order) + (1 | Slant)
## model2_outcome: outcome_confidence_abs ~ ChartType + outcome_aligned + Strength + (1 | Order) + (1 |
## model3_outcome: outcome_confidence_abs ~ ChartType + outcome_aligned + Strength + Prime + (1 | Order)
## model4_outcome: outcome_confidence_abs ~ ChartType + outcome_aligned * Strength + Prime + (1 | Order)
## model5_outcome: outcome_confidence_abs ~ ChartType + outcome_aligned * Strength * Prime + (1 | Order)
## model6_outcome: outcome_confidence_abs ~ ChartType * outcome_aligned * Strength * Prime + (1 | Order)
## model7_outcome: outcome_confidence_abs ~ ChartType * outcome_aligned * Strength * Prime + age + educa
##                 npar    AIC    BIC  logLik deviance   Chisq Df Pr(>Chisq)
## model0_outcome     5 4291.5 4314.0 -2140.8   4281.5
## model1_outcome     7 4287.1 4318.6 -2136.6   4273.1  8.3836  2    0.01512 *
## model2_outcome     8 4288.6 4324.6 -2136.3   4272.6  0.4793  1    0.48873
## model3_outcome     9 4290.5 4331.0 -2136.3   4272.5  0.1060  1    0.74476
## model4_outcome    10 4292.0 4336.9 -2136.0   4272.0  0.5831  1    0.44511
## model5_outcome    14 4293.0 4355.8 -2132.5   4265.0  7.0036  4    0.13570
## model6_outcome    23 4303.0 4406.3 -2128.5   4257.0  7.9429  9    0.53991
## model7_outcome    39 4295.8 4471.0 -2108.9   4217.8 39.1931 16    0.00102 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#### Appraisal ####

model0_author <- lmer(author_confidence_abs ~ ChartType + (1|Order) + (1|Slant), data = df)
```

```
## boundary (singular) fit: see help('isSingular')
```

```r
model1_author <- lmer(author_confidence_abs ~ ChartType + author_aligned + (1|Order) + (1|Slant), data =
```

```
## boundary (singular) fit: see help('isSingular')
```

```r
model2_author <- lmer(author_confidence_abs ~ ChartType + author_aligned + Strength + outcome_aligned +
```

```
## fixed-effect model matrix is rank deficient so dropping 2 columns / coefficients
## boundary (singular) fit: see help('isSingular')
```

```r
model3_author <- lmer(author_confidence_abs ~ ChartType + author_aligned + Strength * outcome_aligned +
```

```
## fixed-effect model matrix is rank deficient so dropping 5 columns / coefficients
## boundary (singular) fit: see help('isSingular')
```

```r
model4_author <- lmer(author_confidence_abs ~ ChartType + author_aligned + Strength * outcome_aligned +
```

```
## fixed-effect model matrix is rank deficient so dropping 5 columns / coefficients
## boundary (singular) fit: see help('isSingular')
```

```
model5_author <- lmer(author_confidence_abs ~ ChartType + author_aligned + Strength * outcome_aligned +
```

```
## fixed-effect model matrix is rank deficient so dropping 5 columns / coefficients
## boundary (singular) fit: see help('isSingular')
```

```
model6_author <- lmer(author_confidence_abs ~ ChartType + author_aligned + Strength * outcome_aligned +
```

```
## fixed-effect model matrix is rank deficient so dropping 5 columns / coefficients
## boundary (singular) fit: see help('isSingular')
```

```
anova(model0_author, model1_author, model2_author, model3_author, model4_author, model5_author, model6_a
```

```
## refitting model(s) with ML (instead of REML)
```

```
## Data: df
## Models:
## model0_author: author_confidence_abs ~ ChartType + (1 | Order) + (1 | Slant)
## model1_author: author_confidence_abs ~ ChartType + author_aligned + (1 | Order) + (1 | Slant)
## model2_author: author_confidence_abs ~ ChartType + author_aligned + Strength + outcome_aligned + (1
## model3_author: author_confidence_abs ~ ChartType + author_aligned + Strength * outcome_aligned + (1
## model4_author: author_confidence_abs ~ ChartType + author_aligned + Strength * outcome_aligned + Pri
## model5_author: author_confidence_abs ~ ChartType + author_aligned + Strength * outcome_aligned + Pri
## model6_author: author_confidence_abs ~ ChartType + author_aligned + Strength * outcome_aligned + Pri
##               npar    AIC    BIC  logLik deviance   Chisq Df Pr(>Chisq)
## model0_author    5 4315.2 4337.6 -2152.6   4305.2
## model1_author    7 4249.6 4281.1 -2117.8   4235.6 69.5591  2  7.860e-16 ***
## model2_author    9 4203.1 4243.5 -2092.5   4185.1 50.5420  2  1.059e-11 ***
## model3_author   10 4201.9 4246.8 -2091.0   4181.9  3.1527  1    0.07580 .
## model4_author   11 4203.2 4252.6 -2090.6   4181.2  0.7444  1    0.38826
## model5_author   13 4207.1 4265.5 -2090.6   4181.1  0.0591  2    0.97090
## model6_author   29 4215.0 4345.2 -2078.5   4157.0 24.1385 16    0.08652 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model2_author)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: author_confidence_abs ~ ChartType + author_aligned + Strength +
##     outcome_aligned + (1 | Order) + (1 | Slant)
##    Data: df
##
## REML criterion at convergence: 4183.4
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7937 -0.5272 -0.3510  0.3649  3.7770
##
## Random effects:
##  Groups   Name        Variance  Std.Dev.
##  Slant    (Intercept) 0.000e+00 0.000e+00
```

```
##  Order    (Intercept) 9.707e-16 3.116e-08
##  Residual             3.352e+01 5.790e+00
## Number of obs: 660, groups:  Slant, 3; Order, 2
##
## Fixed effects:
##                         Estimate Std. Error t value
## (Intercept)               8.0130     0.5215  15.365
## ChartTypeLine             0.3378     0.4509   0.749
## author_alignedNo-Side    -5.2985     0.6092  -8.698
## author_alignedUnaligned  -4.4581     0.6303  -7.073
## StrengthLow              -3.4627     0.5699  -6.076
## outcome_alignedUnaligned  2.0345     0.5621   3.620
##
## Correlation of Fixed Effects:
##             (Intr) ChrtTL at_N-S athr_U StrngL
## ChartTypeLn -0.443
## athr_lgnN-S -0.692  0.009
## athr_lgndUn -0.166  0.010  0.138
## StrengthLow -0.478 -0.002  0.410 -0.248
## otcm_lgndUn -0.451  0.017  0.379 -0.124  0.044
## fit warnings:
## fixed-effect model matrix is rank deficient so dropping 2 columns / coefficients
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

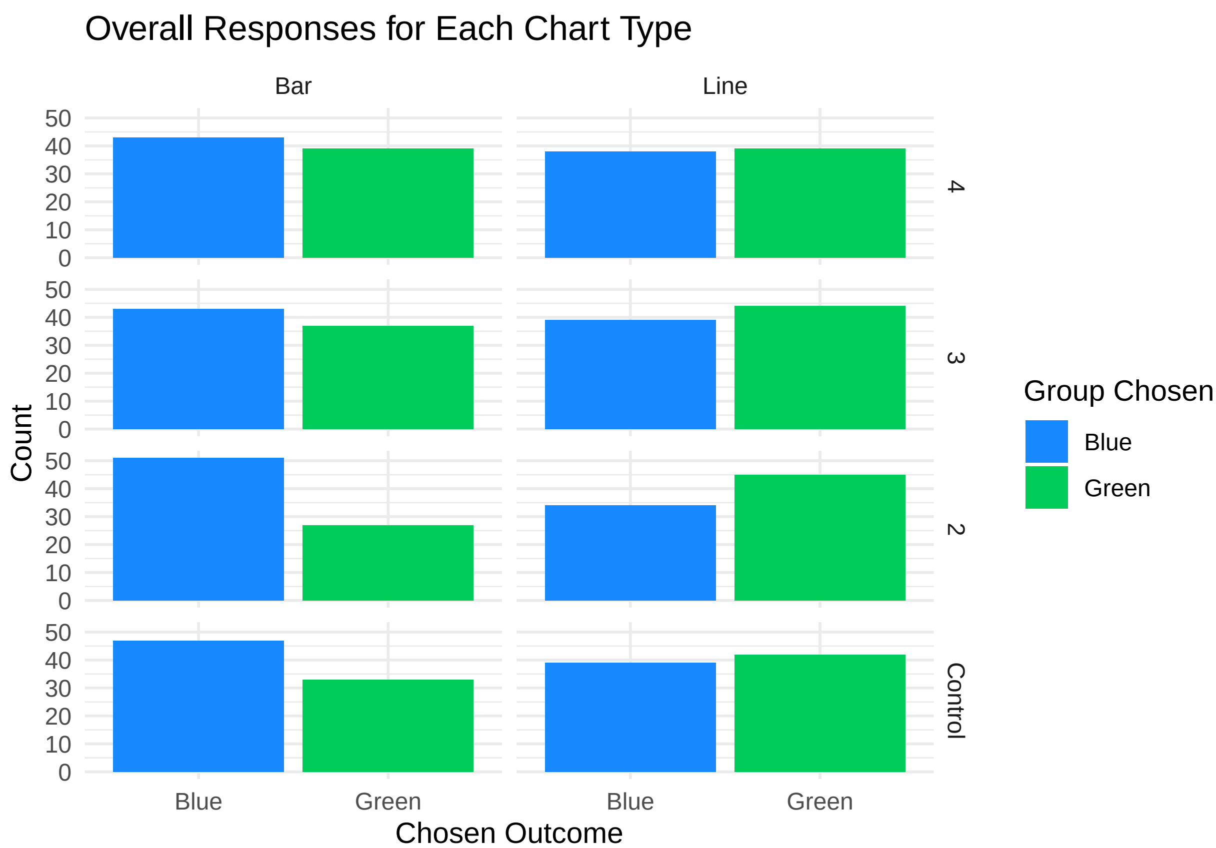
Comparisons to Study 1

Load in study 1 data

```
#### import_and_clean_raw_data ####

# read in raw data
df1 <- read.csv('study 1 data.csv')

#### merging columns ####

# create single column for the outcome chosen by the participant to win (combining the counterbalancing
df1$chosen_outcome <-
  str_remove_all(
    paste0(df1$winning_party, df1$winning_party.1),
    "NA")

# create single column for the confidence rating provided by participants indicating how confident they
df1$outcome_confidence <-
  as.numeric(
    str_remove_all(
      paste0(df1$winning_scale_1, df1$winning_scale_1.1),
      "NA")
  )

# create single column for confidence rating for the likely author of the visualization
df1$author_confidence <-
  as.numeric(
```

```
    str_remove_all(
      paste0(df1$bias_rating_1, df1$bias_rating_1.1),
      "NA")
  )

#### calculating categories ####

# transform data to account for Order counterbalancing
df1$outcome_confidence <- ifelse(
  df1$Order == 'GreenLeft',
  # if the Green responses were on the left (and Blue on the right)
  df1$outcome_confidence * -1,
  # move the Green responses to the right (and Blue to the left)
  df1$outcome_confidence
  # else stay the same
)

df1$author_confidence <- ifelse(
  df1$Order == 'GreenLeft',
  # if the Green responses were on the left (and Blue on the right)
  df1$author_confidence * -1,
  # move the Green responses to the right (and Blue to the left)
  df1$author_confidence
  # else stay the same
)

# create single column for the estimated author, based on the bias rating provided by participants indi
df1$chosen_author <- ifelse(df1$author_confidence < 0,
                            # negative values indicate blue
                            'Blue',
                            ifelse(df1$author_confidence > 0,
                                   # positive values indicate green
                                   'Green',
                                   'Neutral'
                                   # else neutral
                            ))

# create column indicating whether the chosen outcome was aligned with the slant of the text displayed
df1$outcome_aligned <- ifelse(df1$Level != 'Control',
                              # control conditions cannot be aligned or unaligned
                              ifelse(df1$chosen_outcome == 'Blue',
                                     # examine the responses which selected Blue
                                     ifelse(df1$Slant == 'Blue',
                                            # if the slant was Blue, then aligned ("Aligned"), else not
                                            "Aligned", "Unaligned"),
                                     # examine the responses which selected Green
                                     ifelse(df1$Slant == 'Green',
                                            # if the slant was Green, then aligned ("Aligned"), else no
                                            "Aligned", "Unaligned")),
                              # control conditions receive 'N/A'
                              'NA')

# create column indicating whether the chosen author was aligned with the slant of the text displayed
```

```r
df1$author_aligned <- ifelse(df1$Level != 'Control',
                             # control conditions cannot be aligned or unaligned,
                             ifelse(df1$chosen_author == 'Blue',
                                    # examine the responses which selected Blue
                                    ifelse(df1$Slant == 'Blue',
                                           # if the slant was Blue, then aligned ("Aligned"), else not
                                           "Aligned", "Unaligned"),
                                    # else, examine the responses which selected Green
                                    ifelse(df1$Slant == 'Green',
                                           # if the slant was Green, then aligned ("Aligned"), else not
                                           "Aligned", "Unaligned")),
                             # control conditions receive 'N/A'
                             'NA')

# create column with the absolute value of the confidence rating for outcome
df1$outcome_confidence_abs <- abs(df1$outcome_confidence)

# create column with the absolute value of the confidence rating for author
df1$author_confidence_abs <- abs(df1$author_confidence)

# make categorical column to indicate the treatment or control groups
df1$treatment <- ifelse(df1$Level == 'Control', 'Control', 'Treatment')

# make categorical column to indicate the position comparisons
df1$full_condition <- paste(df1$Position, df1$treatment)

# set factor orders
df1$Level <- factor(df1$Level, levels = c(4, 3, 2, "Control"))

## VIS FOR WHICH OPTIONS WERE CHOSEN
ggplot(df1, aes(x = chosen_outcome, fill = chosen_outcome))+
  geom_bar(stat = "count")+
  labs(x = "Chosen Outcome",
       y = "Count",
       title = "Overall Responses for Each Chart Type",
       fill = "Group Chosen")+
  scale_fill_manual(values = c("#1888ff", "#00cd59"))+
  facet_grid(Level~ChartType)
```




```r
df1 %>% group_by(ChartType, chosen_outcome) %>%
  summarise(
    n= n()
  )
```

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

## # A tibble: 4 x 3
## # Groups:   ChartType [2]
##   ChartType chosen_outcome     n
##   <chr>     <chr>          <int>
## 1 Bar       Blue             184
## 2 Bar       Green            136
## 3 Line      Blue             150
## 4 Line      Green            170
```

```r
# select the columns of interest
df1 <-
  select(df1,
         pt,
         Order,
         # assigned value,
         # condition determining the sides of the screen on which each choice option was shown (BlueLef
         Level,
```

```
        # assigned value,
        # condition determining the semantic level of the text shown on the chart (Control, 2, 3, 4)
        Position,
        # assigned value
        # condition determining the position of the text shown on the chart (Title, Annotation)
        full_condition,
        # calculated column,
        # describes the position of the text, along with whether the text was control or a treatment c
        # (Title Control, Title Treatment, Annotation Control, Annotation Treatment)
        Slant,
        # assigned value,
        # condition determining which color the text referred to (Blue, Green)
        ChartType,
        # chart type shown, indicates which experiment the data belong to (Bar, Line)
        treatment,
        # calculated column,
        # indicates whether the participant viewed a control condition or saw text with a slant (Contr
        outcome_confidence_abs,
        # calculated column,
        # absolute value of outcome_confidence response
        outcome_aligned,
        # calculated column,
        # reports whether the outcome selected by a participant was aligned with the text presented to
        author_confidence_abs,
        # calculated column,
        # absolute value of author_confidence response (0, 25)
        author_aligned
        # calculated column,
        # reports whether the author selected by a participant was aligned with the text presented to
  )

#### experiment subset ####

bar1 <- subset(df1, ChartType == 'Bar')
line1 <- subset(df1, ChartType == 'Line')
```

Comparing Values for control and no-side conditions

```
condition_vis1 <- df1 %>%
  group_by(ChartType, outcome_aligned, Level) %>%
  summarize(
    mean_outcome = mean(outcome_confidence_abs),
    sd_outcome = sd(outcome_confidence_abs),
    n_outcome = n(),
    se_outcome = sd_outcome / sqrt(n_outcome),
  ) %>%
  mutate(
    condition = Level,
    study = 1
  ) %>%
  select(
    mean_outcome,
    se_outcome,
```

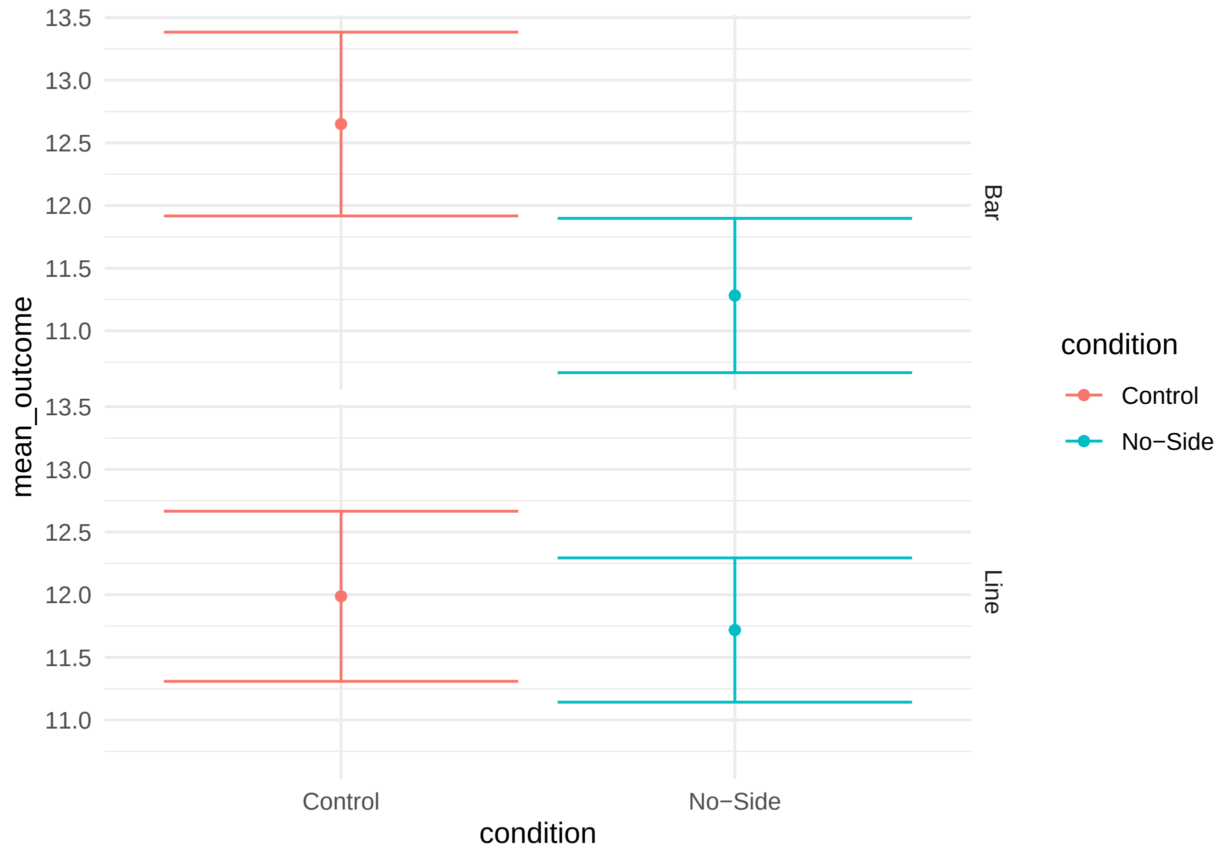
```
    condition,
    study
  )
```

```
## `summarise()` has grouped output by 'ChartType', 'outcome_aligned'. You can
## override using the `.groups` argument.
## Adding missing grouping variables: `ChartType`, `outcome_aligned`
```

```
condition_vis2 <- df %>%
  group_by(ChartType, outcome_aligned, Strength) %>%
  summarize(
    mean_outcome = mean(outcome_confidence_abs),
    sd_outcome = sd(outcome_confidence_abs),
    n_outcome = n(),
    se_outcome = sd_outcome / sqrt(n_outcome)
  ) %>%
  mutate(
    condition = Strength,
    study = 2
  ) %>%
  select (
    mean_outcome,
    se_outcome,
    condition,
    study,
  )
```

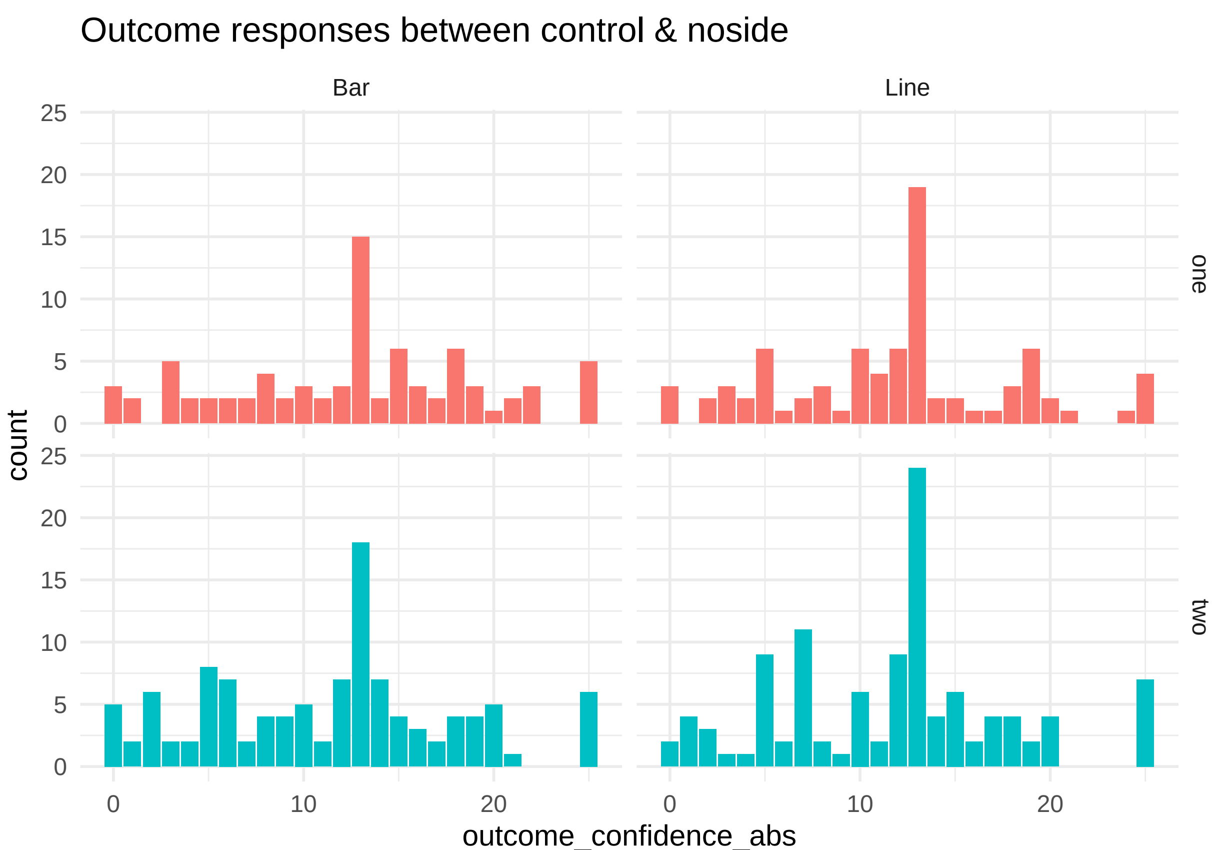
```
## `summarise()` has grouped output by 'ChartType', 'outcome_aligned'. You can
## override using the `.groups` argument.
## Adding missing grouping variables: `ChartType`, `outcome_aligned`
```

```
compare_exp <- rbind(condition_vis2, condition_vis1)

ggplot(subset(compare_exp, condition == "No-Side" | condition == "Control"), aes(x = condition, y = mean
  geom_point()+
  geom_errorbar(aes(ymin = mean_outcome - se_outcome, ymax = mean_outcome + se_outcome))+
  facet_grid(ChartType ~.)
```

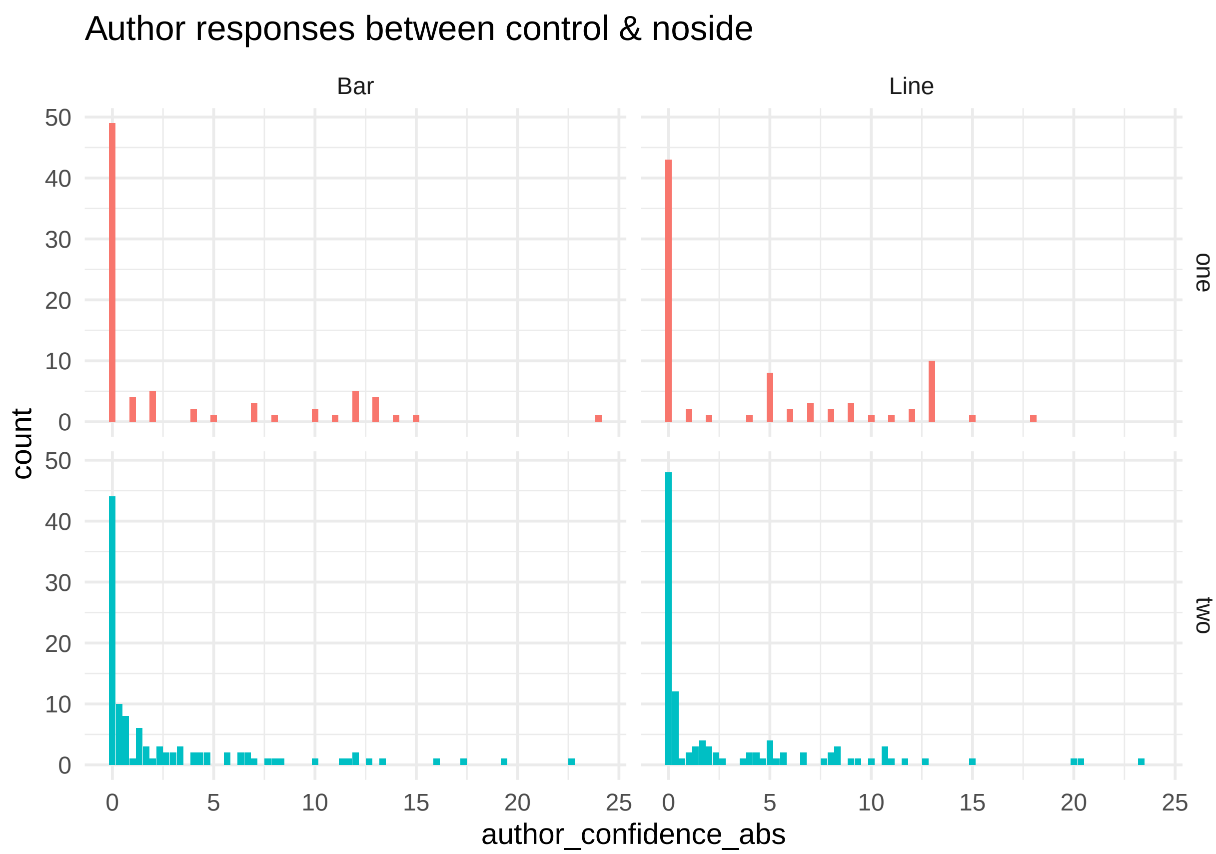



```r
control_responses <- subset(df1, Level == "Control") %>%
  mutate(
    condition = Level,
    study = "one"
  ) %>%
  select(
    pt,
    condition,
    study,
    ChartType,
    outcome_confidence_abs,
    author_confidence_abs)

noside_responses <- subset(df, Strength == "No-Side") %>%
  mutate(
    condition = Strength,
    study = "two"
  ) %>%
  select(
    pt,
    condition,
    study,
    ChartType,
    outcome_confidence_abs,
    author_confidence_abs)
```

```
comp_responses <- rbind(control_responses, noside_responses)

ggplot(comp_responses, aes(x = outcome_confidence_abs, fill = study))+
  geom_bar(stat = "count")+
  facet_grid(study ~ ChartType)+
  theme(legend.position = "none")+
  labs(
    title = "Outcome responses between control & noside"
  )
```




```
ggplot(comp_responses, aes(x = author_confidence_abs, fill = study))+
  geom_bar(stat = "count")+
  facet_grid(study ~ ChartType)+
  theme(legend.position = "none")+
  labs(
    title = "Author responses between control & noside"
  )
```




```r
wilcox.test(outcome_confidence_abs ~ as.factor(study), data = subset(comp_responses, ChartType == "Bar")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  outcome_confidence_abs by as.factor(study)
## W = 4957.5, p-value = 0.1354
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(outcome_confidence_abs ~ as.factor(study), data = subset(comp_responses, ChartType == "Line
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  outcome_confidence_abs by as.factor(study)
## W = 4518, p-value = 0.8676
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(author_confidence_abs ~ as.factor(study), data = subset(comp_responses, ChartType == "Bar"))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  author_confidence_abs by as.factor(study)
```

```
## W = 3887.5, p-value = 0.1453
## alternative hypothesis: true location shift is not equal to 0
```

```r
wilcox.test(author_confidence_abs ~ as.factor(study), data = subset(comp_responses, ChartType == "Line")
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  author_confidence_abs by as.factor(study)
## W = 4593.5, p-value = 0.6986
## alternative hypothesis: true location shift is not equal to 0
```

```r
all_study <- rbind(
  df %>% select(outcome_aligned, outcome_confidence_abs, author_aligned, author_confidence_abs, treatmen
  df1 %>% select(outcome_aligned, outcome_confidence_abs, author_aligned, author_confidence_abs, treatme
)

all_study %>%
  group_by(outcome_aligned) %>%
  summarise(
    n = n()
  )
```

```
## # A tibble: 4 x 2
##   outcome_aligned     n
##   <chr>           <int>
## 1 Aligned           509
## 2 NA                161
## 3 No-Side           220
## 4 Unaligned         410
```

```r
dunnTest(mean ~ as.factor(outcome_aligned), data = subset(all_study, treatment == "Treatment") %>%
           group_by(outcome_aligned) %>%
           summarise(
             mean = mean(outcome_confidence_abs)
           ), method = "bonferroni")
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison

##   p-values adjusted with the Bonferroni method.

##            Comparison Z   P.unadj     P.adj
## 1 Aligned - Unaligned 1 0.3173105 0.3173105
```

```r
subset(all_study, treatment == "Treatment") %>%
  group_by(outcome_aligned) %>%
  summarise(
    mean = mean(outcome_confidence_abs)
  )
```

```
## # A tibble: 2 x 2
##   outcome_aligned  mean
##   <chr>           <dbl>
## 1 Aligned          13.2
## 2 Unaligned        11.5
```

```r
all_study %>%
  group_by(author_aligned) %>%
  summarise(
    n = n()
  )
```

```
## # A tibble: 4 x 2
##   author_aligned     n
##   <chr>          <int>
## 1 Aligned          621
## 2 NA               161
## 3 No-Side          220
## 4 Unaligned        298
```

```r
dunnTest(mean ~ as.factor(author_aligned), data = subset(all_study, treatment == "Treatment") %>%
           group_by(author_aligned) %>%
           summarise(
             mean = mean(author_confidence_abs)
           ), method = "bonferroni")
```

```
## Dunn (1964) Kruskal-Wallis multiple comparison
##   p-values adjusted with the Bonferroni method.

##            Comparison Z   P.unadj     P.adj
## 1 Aligned - Unaligned 1 0.3173105 0.3173105
```

```r
all_study %>%
  group_by(author_aligned) %>%
  summarise(
    mean = mean(author_confidence_abs)
  )
```

```
## # A tibble: 4 x 2
##   author_aligned  mean
##   <chr>          <dbl>
## 1 Aligned         7.88
## 2 NA              3.63
## 3 No-Side         2.88
## 4 Unaligned       2.89
```

```r
df %>%
  group_by(outcome_aligned) %>%
  summarise(
    mean_chosen = mean(pct_chosen),
    mean_unchosen = mean(pct_unchosen),
    mean_tie = mean(pct_tie)
  )
```

```
## # A tibble: 3 x 4
##   outcome_aligned mean_chosen mean_unchosen mean_tie
##   <chr>                 <dbl>         <dbl>    <dbl>
## 1 Aligned                59.1          28.9     11.9
## 2 No-Side                53.5          30.8     15.7
## 3 Unaligned              54.9          31.6     13.5
```

```
df %>%
  group_by(author_aligned) %>%
  summarise(
    mean_toward = mean(abs(biasToward)),
    mean_against = mean(abs(-1 * biasAgainst)),
    mean_personal = mean(abs(personalView))
  )
```

```
## # A tibble: 3 x 4
##   author_aligned mean_toward mean_against mean_personal
##   <chr>                <dbl>        <dbl>         <dbl>
## 1 Aligned               8.87         7.11          9.53
## 2 No-Side               3.66         3.13          3.96
## 3 Unaligned             3.28         2.65          3.14
```