Statistical analysis brainstorm

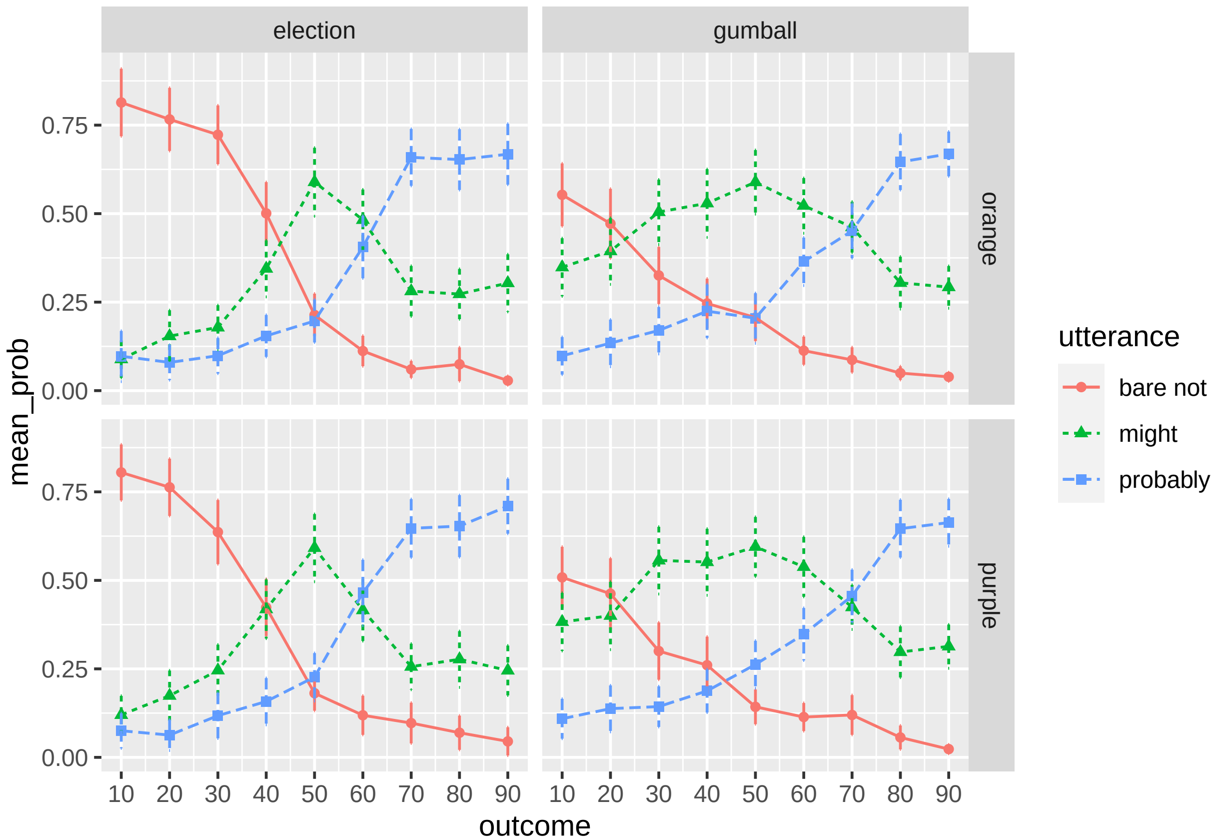
Load data

```
dat = read.csv('../data/processed_data.csv')
```

Plotting data

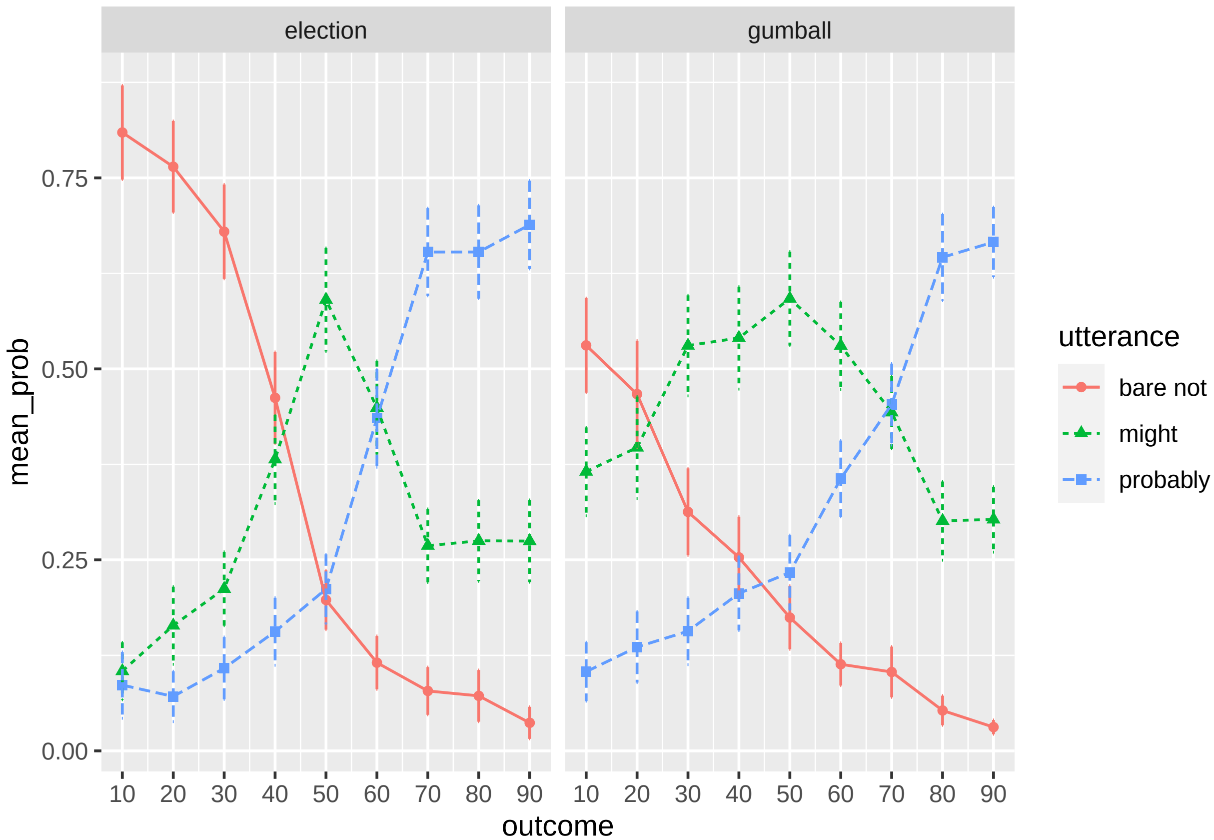
```
summ1 <- dat %>%
  group_by(event_type, outcome, color, utterance) %>%
  summarize(mean_prob = mean(prob),
            se_prob = sd(prob)/sqrt(n()),
            .groups='drop')


ggplot(summ1, aes(x=outcome, y=mean_prob,
                 color=utterance, linetype = utterance, shape=utterance)) +
  geom_point() +
  geom_line() +
  geom_errorbar(aes(ymin=mean_prob - 2*se_prob,
                    ymax=mean_prob + 2*se_prob),
                width = 0.2) +
  facet_grid(color~event_type) +
  scale_x_continuous(breaks=seq(10, 90, 10))
```




```r
summ2 <- dat %>%
  group_by(event_type, outcome, utterance) %>%
  summarize(mean_prob = mean(prob),
            se_prob = sd(prob)/sqrt(n()),
            .groups='drop')


ggplot(summ2, aes(x=outcome, y=mean_prob,
                  color=utterance, linetype = utterance, shape=utterance)) +
  geom_point() +
  geom_line() +
  geom_errorbar(aes(ymin=mean_prob - 2*se_prob,
                    ymax=mean_prob + 2*se_prob),
                width = 0.2) +
  facet_grid(~event_type) +
  scale_x_continuous(breaks=seq(10, 90, 10))
```



Ideas for analysis

Goal: we want to know if participants are using uncertainty expressions differently for election vs. gumball cases. Specifically, it looks like they are using "might" and "bare not" very differently. We want to see if this is a significant difference.

Option 1: Use logistic regression The goal of logistic regression would be to fit the curve. The issue with this approach is that "might" seems to have a u-shaped curve that cannot be fit with logistic regression. One way to counter this is to split the data into two parts: outcomes less than 50% and greater than 50%.

Why 50%?

- Empirically it looks like that is where the curve changes (both in our data, but also in Experiment 1 of Schuster and Degen).
- Conceptually it makes sense that at 50% people's perception of events changes?

Option 1a Use logistic regression separately for each utterance (prob of utterance, and 1-prob of utterance).

Pro: Easy to interpret. For HSP abstract for example, easier to make the point about P(might) – because really looks like P(probably) isn't really that different.

Con: Makes an independent assumption thats not fully valid?

Option 1b

Use multinomial logistic regression.

Pro: Seems like the "right" way to model this data.

Con: I am not fully sure how to interpret these models.

Option 2: Use area under the curve Schuster and Degen use this approach to model their data.

"Following Yildirim et al., 2016, we quantified this prediction by fitting a spline with four knots for each expression and each participant and computing the area under the curve (AUC) for the splines corresponding to each expression and participant. The area under the curve is proportional to how highly and for how large a range of event probabilities participants rate an utterance. If an utterance is rated highly for a larger range of event probabilities, the AUC will also be larger. We therefore tested whether listeners updated their expectations according to these intuitions by computing the difference between the AUC of the spline for MIGHT and of the spline for PROBABLY for each participant. We predicted that the mean AUC difference would be larger in the cautious speaker condition than in the confident speaker condition."

Pro: Does not require us to split the data in half. Also can say "as in previous work". Overall seems like a reasonable way of modeling the data.

Con: I am not 100% sure how to do this/ interpret this correctly under time crunch. Also worried that it might be tricky to explain in HSP abstract.

Decision for HSP abstract Go with Option 1a because it seems the most straightforward to interpret and explain (even if not perfect). Can state in future work that other analysis methods should be used?

Also deciding to average over the different colors because we do not expect any differences there. Maybe treat this as a random effect?

Logistic regression for "might"

First half Predictions:

1. Positive coefficient for `outcome` (when it is z-scored): as P(outcome) increases from 10 to 50, people use might more frequently.
2. Positive coefficient for `eventtype` (where gumball is coded as 1): people use might more for gumball than for election
3. Negative coefficient for `eventtype:gumball`: as P(outcome) approaches 50%, people's use of "might" increases more for election than gumball (since gumball will be high from the beginning).

```
might_firsthalf <- glmer(prob ~ scale(outcome)*event_type +
                           (1 + scale(outcome)*event_type | random_id) +
                           (1 + scale(outcome)*event_type | color),
                         family = binomial,
                         data=subset(dat, outcome <= 50
                                     & utterance=="might"))
```

```
## Warning in eval(family$initialize, rho): non-integer #successes in a binomial
## glm!

## Warning in (function (fn, par, lower = rep.int(-Inf, n), upper = rep.int(Inf, :
## failure to converge in 10000 evaluations

## Warning in optwrap(optimizer, devfun, start, rho$lower, control = control, :
## convergence code 4 from Nelder_Mead: failure to converge in 10000 evaluations

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

```
summary(might_firsthalf)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
```

```
## prob ~ scale(outcome) * event_type + (1 + scale(outcome) * event_type |
##     random_id) + (1 + scale(outcome) * event_type | color)
##    Data: subset(dat, outcome <= 50 & utterance == "might")
##
##      AIC      BIC   logLik deviance df.resid
##    875.6    993.5   -413.8    827.6      984
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -11.2437  -0.1474   0.1139   0.4517  12.6579
##
## Random effects:
##  Groups    Name                             Variance Std.Dev. Corr
##  random_id (Intercept)                      0.69580  0.8341
##            scale(outcome)                   3.28000  1.8111   -0.84
##            event_typegumball                4.51828  2.1256   -0.55  0.91
##            scale(outcome):event_typegumball 1.69847  1.3033    0.55 -0.91 -1.00
##  color     (Intercept)                      0.00000  0.0000
##            scale(outcome)                   0.03084  0.1756     NaN
##            event_typegumball                0.09110  0.3018     NaN  0.93
##            scale(outcome):event_typegumball 0.05471  0.2339     NaN -1.00 -0.92
## Number of obs: 1008, groups:  random_id, 24; color, 2
##
## Fixed effects:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -2.9527     0.3384  -8.727  < 2e-16 ***
## scale(outcome)                     1.8816     0.4639   4.056 4.99e-05 ***
## event_typegumball                  2.5434     0.5631   4.517 6.29e-06 ***
## scale(outcome):event_typegumball  -1.4652     0.4286  -3.419 0.000629 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) scl(t) evnt_t
## scale(otcm) -0.746
## evnt_typgmb -0.681  0.841
## scl(tcm):v_  0.658 -0.883 -0.823
## optimizer (Nelder_Mead) convergence code: 4 (failure to converge in 10000 evaluations)
## boundary (singular) fit: see help('isSingular')
## failure to converge in 10000 evaluations
```

Second half Predictions:

1. Negative coefficient for `outcome` (when it is z-scored): as P(outcome) increases from 50 to 100, people use might less frequently.
2. Positive coefficient for `eventtype` (where gumball is coded as 1): people use might more for gumball than for election
3. Positive coefficient for `eventtype:gumball`: as P(outcome) approaches 100%, people's use of "might" is higher for gumball than for election. (might is still broadly acceptable for gumball).

```
might_secondhalf <- glmer(prob ~ scale(outcome)*event_type +
                           (1 + scale(outcome)*event_type | random_id) +
                           (1 + scale(outcome)*event_type | color),
                         family = binomial,
```

```r
                         data=subset(dat, outcome > 50
                                     & utterance=="might"))
```

```
## Warning in eval(family$initialize, rho): non-integer #successes in a binomial
## glm!

## Warning in (function (fn, par, lower = rep.int(-Inf, n), upper = rep.int(Inf, :
## failure to converge in 10000 evaluations

## Warning in optwrap(optimizer, devfun, start, rho$lower, control = control, :
## convergence code 4 from Nelder_Mead: failure to converge in 10000 evaluations

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues
```

```r
summary(might_secondhalf)
```

```
## Warning in vcov.merMod(object, use.hessian = use.hessian): variance-covariance matrix computed from f
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## Warning in vcov.merMod(object, correlation = correlation, sigm = sig): variance-covariance matrix com
## not positive definite or contains NA values: falling back to var-cov estimated from RX

## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## prob ~ scale(outcome) * event_type + (1 + scale(outcome) * event_type |
##     random_id) + (1 + scale(outcome) * event_type | color)
##    Data: subset(dat, outcome > 50 & utterance == "might")
##
##      AIC      BIC   logLik deviance df.resid
##    757.5    870.4   -354.8    709.5      792
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.0812 -0.1246  0.5759  0.8664  4.2066
##
## Random effects:
##  Groups    Name                           Variance Std.Dev. Corr
##  random_id (Intercept)                    0.175376 0.41878
##            scale(outcome)                 0.631978 0.79497   0.61
##            event_typegumball              0.640392 0.80025  -1.00 -0.64
##            scale(outcome):event_typegumball 0.297527 0.54546   0.58 -0.27 -0.54
##  color     (Intercept)                    0.008986 0.09480
##            scale(outcome)                 0.005189 0.07203   0.46
##            event_typegumball              0.030468 0.17455  -1.00 -0.47
##            scale(outcome):event_typegumball 0.027366 0.16543  -0.65 -0.63  0.66
## Number of obs: 816, groups:  random_id, 24; color, 2
##
## Fixed effects:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -2.1639     0.2084 -10.381  < 2e-16 ***
## scale(outcome)                    -0.7795     0.2397  -3.251  0.00115 **
```

```
## event_typegumball                  0.7961     0.2975   2.676  0.00746 **
## scale(outcome):event_typegumball   0.4114     0.2668   1.542  0.12311
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) scl(t) evnt_t
## scale(otcm)  0.500
## evnt_typgmb -0.857 -0.482
## scl(tcm):v_ -0.268 -0.584  0.194
## optimizer (Nelder_Mead) convergence code: 4 (failure to converge in 10000 evaluations)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
## failure to converge in 10000 evaluations
```

Logistic regression for "bare not"

Predictions inverse of might.

1. Negative coefficient for `outcome` (when it is z-scored): as P(outcome) increases from 10 to 50, people use not less frequently.
2. Negatie coefficient for `eventtype` (where gumball is coded as 1): people use bare not less for gumball than for election
3. Positive coefficient for `eventtype:gumball`: as P(outcome) approaches 50%, people's use of "bare not" decreases more for election than gumball (two negatives make positive).

```
not_firsthalf <- glmer(prob ~ scale(outcome)*event_type +
                           (1 + scale(outcome)*event_type | random_id) +
                           (1 + scale(outcome)*event_type | color),
                         family = binomial,
                         data=subset(dat, outcome <= 50
                                     & utterance=="bare not"))
```

```
## Warning in eval(family$initialize, rho): non-integer #successes in a binomial
## glm!

## Warning in (function (fn, par, lower = rep.int(-Inf, n), upper = rep.int(Inf, :
## failure to converge in 10000 evaluations

## Warning in optwrap(optimizer, devfun, start, rho$lower, control = control, :
## convergence code 4 from Nelder_Mead: failure to converge in 10000 evaluations

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## unable to evaluate scaled gradient

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge: degenerate Hessian with 1 negative eigenvalues
```

```
summary(not_firsthalf)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## prob ~ scale(outcome) * event_type + (1 + scale(outcome) * event_type |
##     random_id) + (1 + scale(outcome) * event_type | color)
##    Data: subset(dat, outcome <= 50 & utterance == "bare not")
```

```
##
##      AIC      BIC   logLik deviance df.resid
##    857.9    975.9   -405.0    809.9      984
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -9.4978 -0.3593 -0.0509  0.3259  5.2486
##
## Random effects:
##  Groups    Name                             Variance Std.Dev. Corr
##  random_id (Intercept)                      1.482572 1.21761
##            scale(outcome)                   0.339159 0.58237  -0.99
##            event_typegumball                3.124330 1.76758   0.18 -0.10
##            scale(outcome):event_typegumball 1.559020 1.24861   0.92 -0.96 -0.20
##  color     (Intercept)                      0.009974 0.09987
##            scale(outcome)                   0.013841 0.11765   0.53
##            event_typegumball                0.019773 0.14062   0.66  0.98
##            scale(outcome):event_typegumball 0.058805 0.24250  -0.49 -1.00 -0.98
## Number of obs: 1008, groups:  random_id, 24; color, 2
##
## Fixed effects:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                        1.5061     0.3268   4.608 4.06e-06 ***
## scale(outcome)                    -2.4677     0.2720  -9.074  < 2e-16 ***
## event_typegumball                 -2.8099     0.4698  -5.981 2.21e-09 ***
## scale(outcome):event_typegumball   0.9066     0.4161   2.179   0.0294 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) scl(t) evnt_t
## scale(otcm) -0.634
## evnt_typgmb -0.146  0.300
## scl(tcm):v_  0.564 -0.810 -0.259
## optimizer (Nelder_Mead) convergence code: 4 (failure to converge in 10000 evaluations)
## unable to evaluate scaled gradient
## Model failed to converge: degenerate  Hessian with 1 negative eigenvalues
## failure to converge in 10000 evaluations
```