Analysis for Wh + Distance

```
rm(list = ls())
library(tidyverse)
```

```
## -- Attaching packages ---------------------------------------------------------------------------------
```

```
## √ ggplot2 2.2.1     √ purrr   0.2.4
## √ tibble  1.4.2     √ dplyr   0.7.5
## √ tidyr   0.8.1     √ stringr 1.3.1
## √ readr   1.1.1     √ forcats 0.3.0
```

```
## -- Conflicts ------------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(brms)
```

```
## Loading required package: Rcpp
```

```
## Loading 'brms' package (version 2.3.0). Useful instructions
## can be found by typing help('brms'). A more detailed introduction
## to the package is available through vignette('brms_overview').
## Run theme_set(theme_default()) to use the default bayesplot theme.
```

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
##
## Attaching package: 'lme4'
```

```
## The following object is masked from 'package:brms':
##
##     ngrps
```

```
library(lmerTest)
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##     lmer
```

```
## The following object is masked from 'package:stats':
##
##     step
```

```
library(plotrix)
library(stringr)
library(readxl)
```

```r
remove_na = function(x) {
  x[!is.na(x)]
}

d = read_csv("tests/combined_results.csv") %>%
  select(-1, -2) %>%
  mutate(unk=unk == "True") %>%
  mutate(region=if_else(region=="prefix" | region=="obj wh" | region=="goal wh" | region=="that", "prefi
  mutate(region=if_else(region=="short modifier" | region=="medium modifier" | region=="long modifier",
  separate(condition, sep="_", into=c("wh", "gap", "gap_position", "modifier"))
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   `Unnamed: 0` = col_integer(),
##   sent_index = col_integer(),
##   word_index = col_integer(),
##   word = col_character(),
##   region = col_character(),
##   condition = col_character(),
##   model_word = col_character(),
##   surprisal = col_double(),
##   model = col_character(),
##   unk = col_character()
## )
```

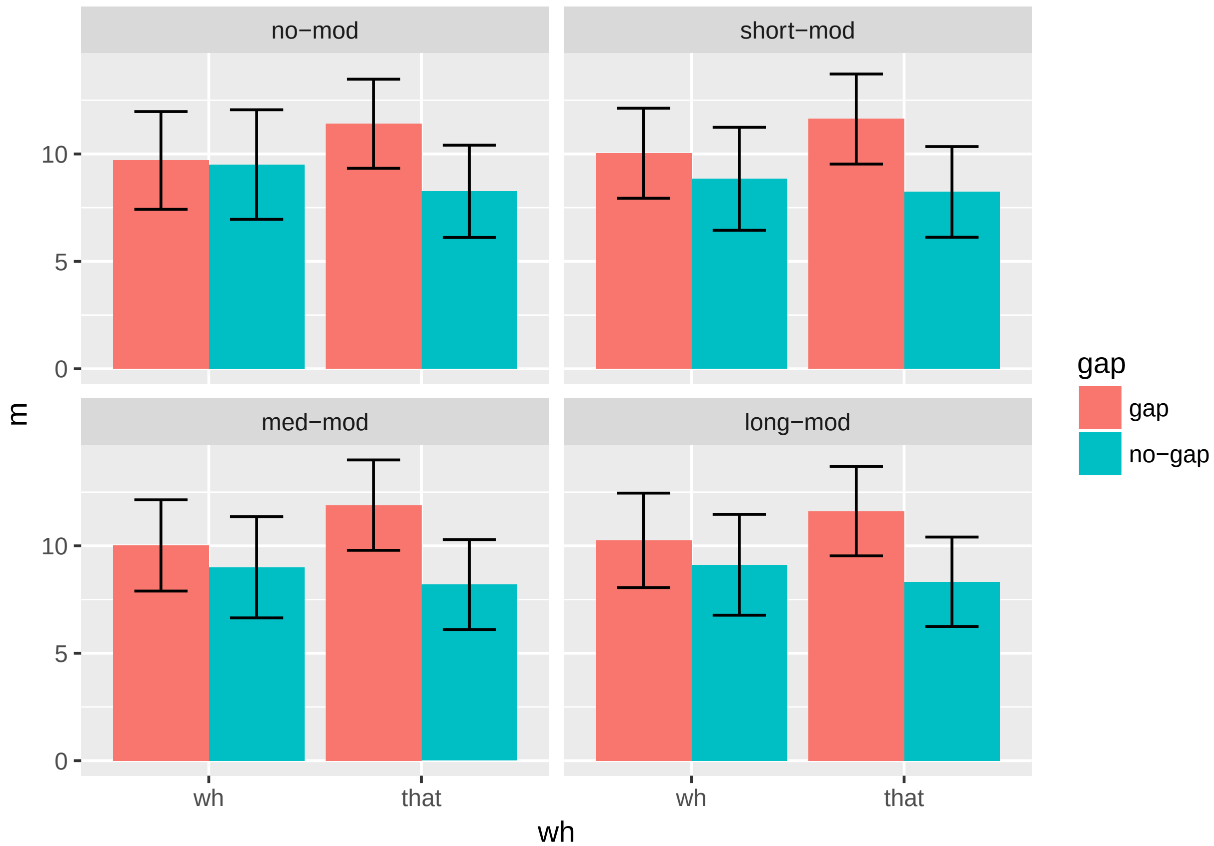
```r
d_agg = d %>%
  group_by(model, region, sent_index, wh, gap, gap_position, modifier) %>%
    summarise(surprisal=sum(surprisal),
              unk=any(unk)) %>%
    ungroup() %>%
  filter(!unk) %>%
  mutate(wh_numeric=if_else(wh == "wh", 1, -1),
         wh=factor(wh, levels=c("wh", "that")),
         gap=factor(gap, levels=c("gap", "no-gap")),
         gap_position=factor(gap_position, levels=c("obj", "goal")),
         modifier=factor(modifier, levels=c("no-mod", "short-mod", "med-mod", "long-mod")))
```

Analysis 1: Gap in object position

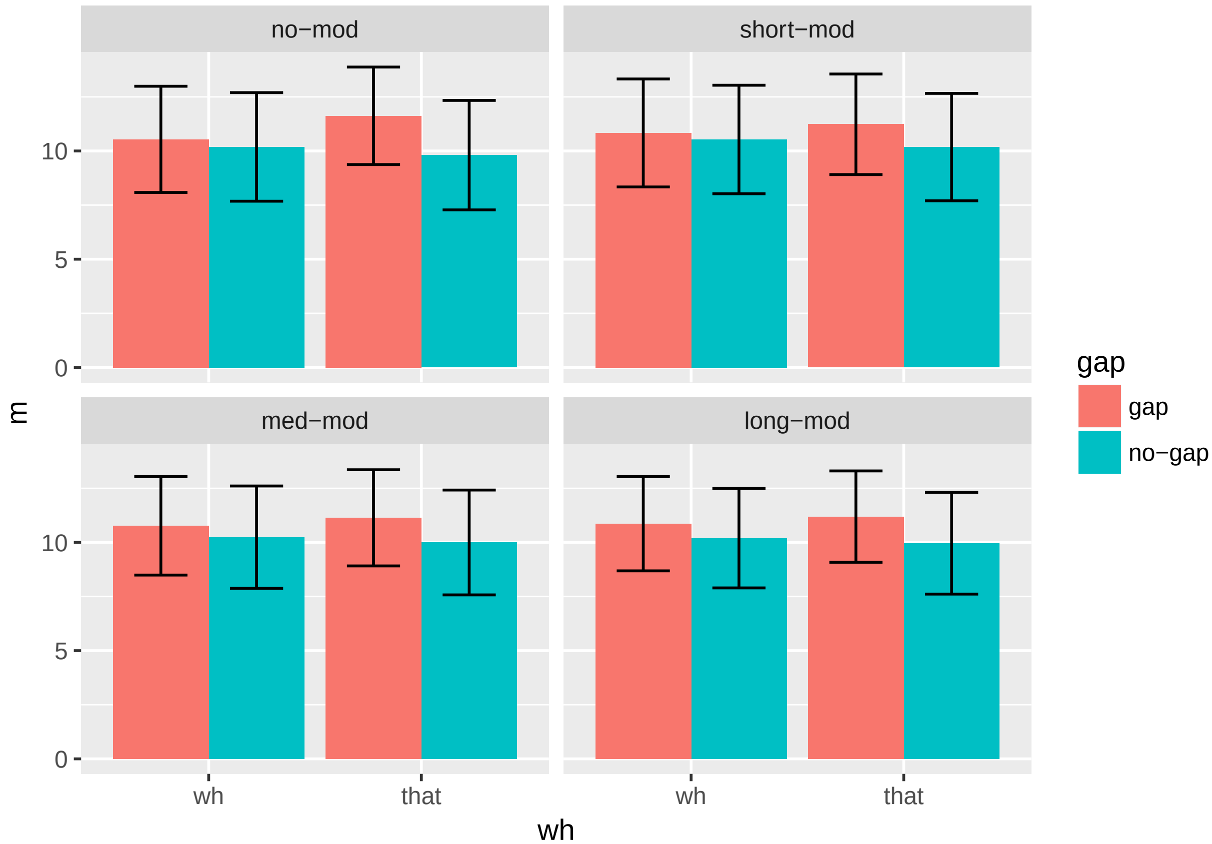
Okay, let's do a quick visualization to see what's going on here.

```r
d2 = d_agg %>%
  filter(model=="google") %>%
  filter(gap_position=="obj") %>%
  filter(region == "to" | region=="goal") %>%
  group_by(model, wh, gap, modifier) %>%
    summarise(m=mean(surprisal),
              s=std.error(surprisal),
              upper=m + 1.96*s,
              lower=m - 1.96*s) %>%
    ungroup()
```

```r
ggplot(d2, aes(x=wh, y=m, ymin=lower, ymax=upper, fill=gap)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~modifier)
```

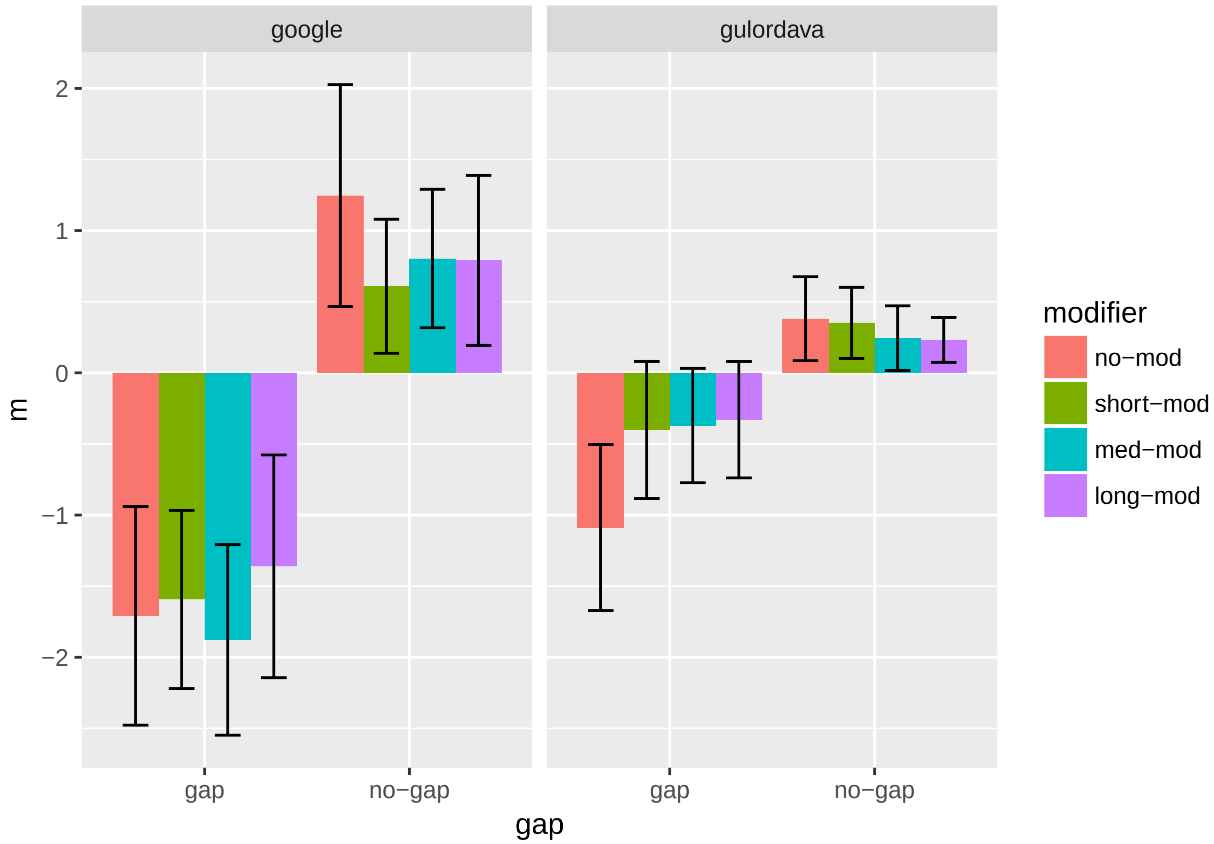



```r
d2 = d_agg %>%
  filter(model=="gulordava") %>%
  filter(gap_position=="obj") %>%
  filter(region == "to" | region=="goal") %>%
  group_by(model, wh, gap, modifier) %>%
    summarise(m=mean(surprisal),
              s=std.error(surprisal),
              upper=m + 1.96*s,
              lower=m - 1.96*s) %>%
    ungroup()

ggplot(d2, aes(x=wh, y=m, ymin=lower, ymax=upper, fill=gap)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~modifier)
```



Great, this is good evidence for filler effects in the google model. We see that with no modifiers, there is higher surprisal in the that/gap condition compared to the that/no-gap condition, but about equal surprisal when there is a wh-licensor. Again, the model is learning half of the dependency. However, with the inclusion of intervening material, the wh condition starts to look more and more like the "that" condition, with lower surprisal in the no-gap case than when a gap is present. It seems as if inclusion of intervening material resets the network, making it "forget" that a gap has been licensed by a wh-word earlier in the sentence.

```
d_wh_effect = d_agg %>%
  filter(region == "to" | region =="goal") %>%
  filter(gap_position=="obj") %>%
  select(-wh_numeric) %>%
  spread(wh, surprisal) %>%
  mutate(wh_effect=wh-`that`)

d_wh_effect %>%
  group_by(model, gap, gap_position, modifier) %>%
    summarise(m=mean(wh_effect),
              s=std.error(wh_effect),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
  ggplot(aes(x=gap, y=m, ymin=lower, ymax=upper, fill=modifier)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model)
```

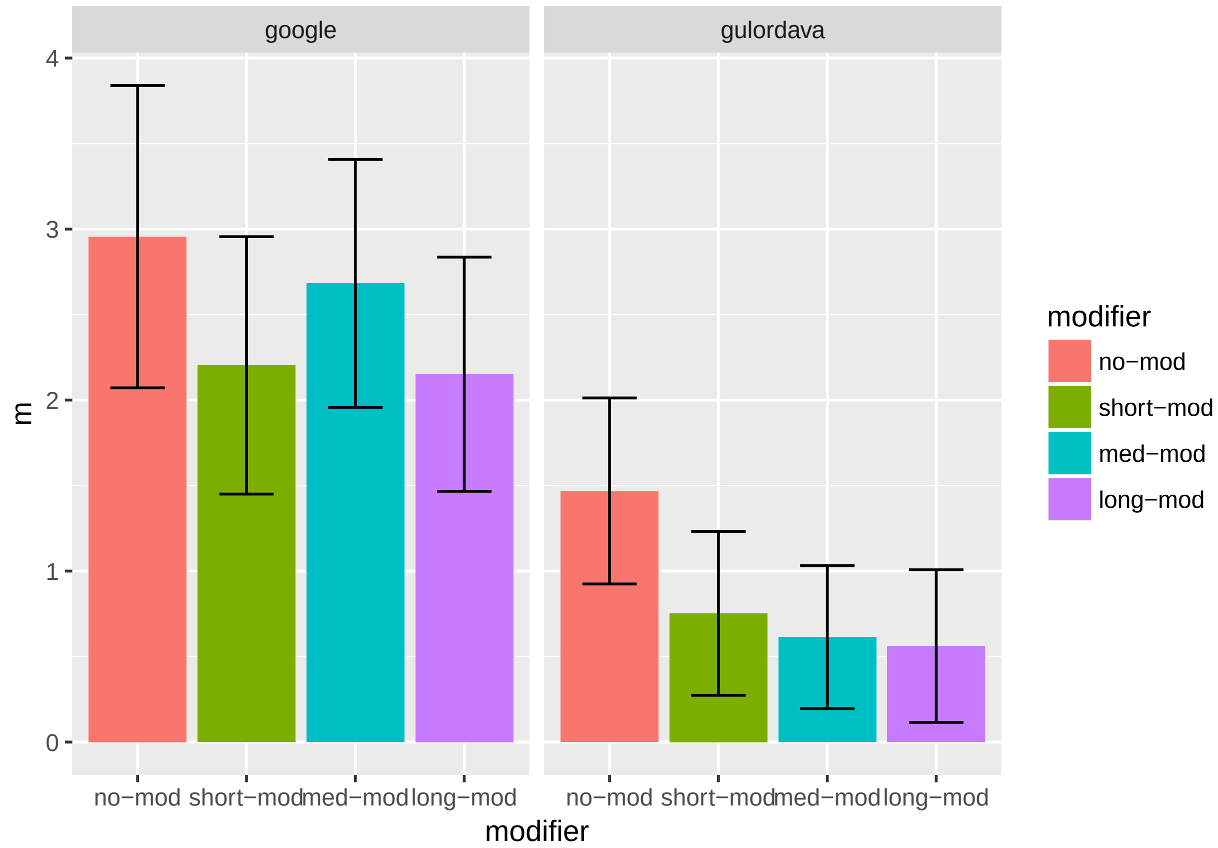


As we would expect, there is much less surprsial when the gap is licensed by a wh-word as opposed to "that". It seems that for the Gulordava model there is an effect of distance (i.e. the gap between the two gets smaller the more mateiral is added), although we can't really tell if it is significant from the graph. For the google model it looks as if the results are more mixed. While it's obvious the no-gap condition is going to show the biggest reduction in surprisal, it's not obvious that the long-modifier condition is any worse than the medium modifier condition.

Let's plot the difference:

```
d_full_interaction = d_agg %>%
  filter(region == "to" | region == "goal") %>%
  filter(gap_position=="obj") %>%
  select(-wh_numeric) %>%
  spread(gap, surprisal) %>%
  mutate(gap_effect=`no-gap`-gap) %>%
  select(-unk, -gap, -`no-gap`) %>%
  spread(wh, gap_effect) %>%
  mutate(wh_interaction=wh-`that`)

d_full_interaction %>%
  group_by(model, modifier) %>%
    summarise(m=mean(wh_interaction, na.rm=T),
              s=std.error(wh_interaction, na.rm=T),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
  ggplot(aes(x=modifier, y=m, ymin=lower, ymax=upper, fill=modifier)) +
```

```
    geom_bar(stat="identity") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model)
```



Statistics:

```
m_google = d_agg %>%
  filter(model == "google", region == "to" | region=="goal", gap_position=="obj") %>%
  lmer(surprisal ~ gap * wh_numeric * modifier +
                  (gap+wh_numeric+modifier|sent_index),
                data=.)
summary(m_google)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: surprisal ~ gap * wh_numeric * modifier + (gap + wh_numeric +
##     modifier | sent_index)
##    Data: .
##
## REML criterion at convergence: 4492.2
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.92910 -0.79408 -0.04529  0.70957  2.70039
##
## Random effects:
##  Groups     Name              Variance  Std.Dev. Corr
```

```
##  sent_index (Intercept)        7.480096 2.73498
##             gapno-gap          0.173167 0.41613  -1.00
##             wh_numeric         0.011017 0.10496  -1.00  1.00
##             modifiershort-mod  0.001217 0.03488   1.00 -1.00 -1.00
##             modifiermed-mod    0.025672 0.16022  -1.00  1.00  1.00 -1.00
##             modifierlong-mod   0.080225 0.28324  -1.00  1.00  1.00 -1.00
##  Residual                     47.117109 6.86419
##
##
##
##
##
##
##   1.00
##
## Number of obs: 672, groups:  sent_index, 21
##
## Fixed effects:
##                                         Estimate Std. Error        df
## (Intercept)                             10.55129    0.95766  42.52305
## gapno-gap                               -1.66809    1.06305 575.54727
## wh_numeric                              -0.85453    0.74929 632.67373
## modifiershort-mod                        0.27882    1.05919 635.02961
## modifiermed-mod                          0.40609    1.05974 616.03068
## modifierlong-mod                         0.38541    1.06097 577.59587
## gapno-gap:wh_numeric                     1.47764    1.05917 636.00000
## gapno-gap:modifiershort-mod             -0.62353    1.49789 636.00000
## gapno-gap:modifiermed-mod               -0.68951    1.49789 636.00000
## gapno-gap:modifierlong-mod              -0.54407    1.49789 636.00000
## wh_numeric:modifiershort-mod             0.05799    1.05917 636.00000
## wh_numeric:modifiermed-mod              -0.08473    1.05917 636.00000
## wh_numeric:modifierlong-mod              0.17424    1.05917 636.00000
## gapno-gap:wh_numeric:modifiershort-mod  -0.37631    1.49789 636.00000
## gapno-gap:wh_numeric:modifiermed-mod    -0.13647    1.49789 636.00000
## gapno-gap:wh_numeric:modifierlong-mod   -0.40190    1.49789 636.00000
##                                        t value Pr(>|t|)
## (Intercept)                             11.018 4.87e-14 ***
## gapno-gap                               -1.569    0.117
## wh_numeric                              -1.140    0.255
## modifiershort-mod                        0.263    0.792
## modifiermed-mod                          0.383    0.702
## modifierlong-mod                         0.363    0.717
## gapno-gap:wh_numeric                     1.395    0.163
## gapno-gap:modifiershort-mod             -0.416    0.677
## gapno-gap:modifiermed-mod               -0.460    0.645
## gapno-gap:modifierlong-mod              -0.363    0.717
## wh_numeric:modifiershort-mod             0.055    0.956
## wh_numeric:modifiermed-mod              -0.080    0.936
## wh_numeric:modifierlong-mod              0.165    0.869
## gapno-gap:wh_numeric:modifiershort-mod  -0.251    0.802
## gapno-gap:wh_numeric:modifiermed-mod    -0.091    0.927
## gapno-gap:wh_numeric:modifierlong-mod   -0.268    0.789
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
```

```r
m_gul = d_agg %>%
  filter(model == "gulordava", region == "to" | region=="goal", gap_position=="obj") %>%
  lmer(surprisal ~ gap * wh_numeric * modifier+
                 (gap+wh_numeric+modifier|sent_index),
               data=.)
summary(m_gul)
```

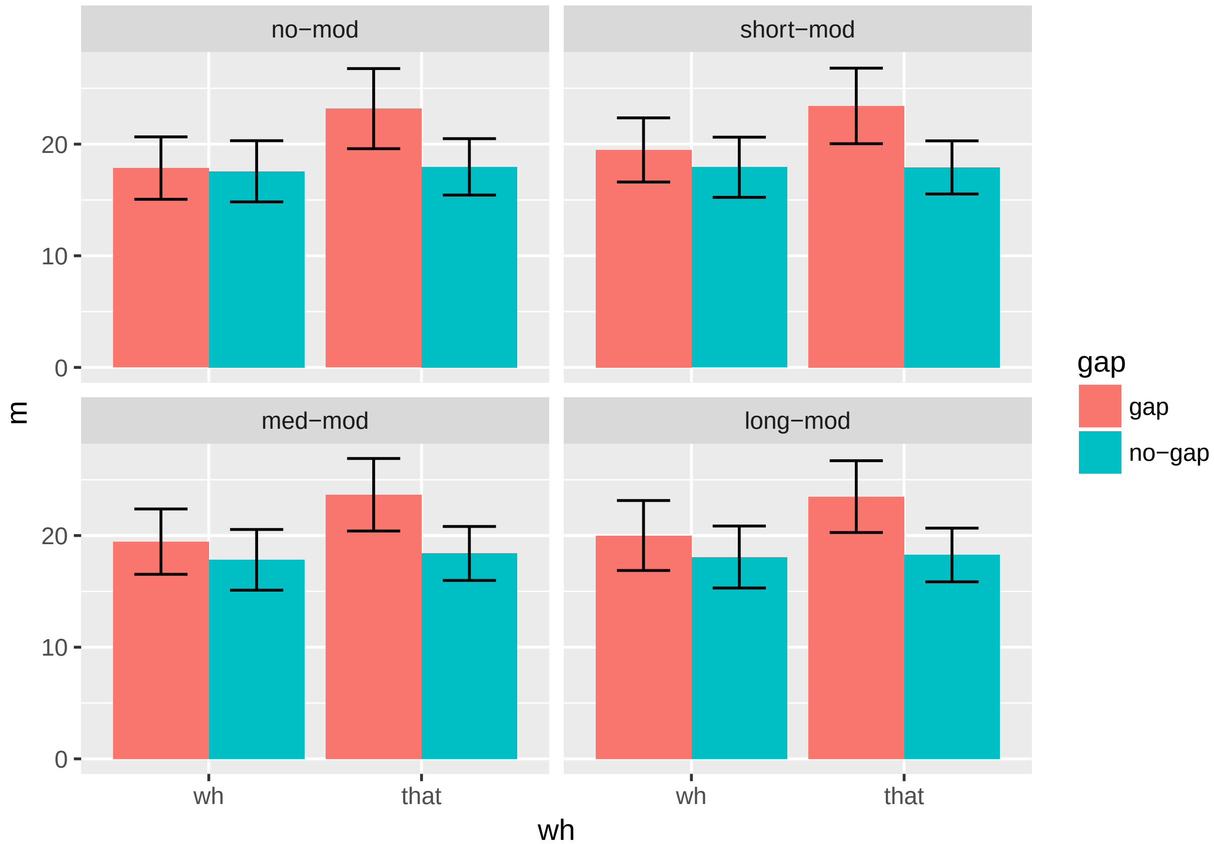
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: surprisal ~ gap * wh_numeric * modifier + (gap + wh_numeric +
##     modifier | sent_index)
##    Data: .
##
## REML criterion at convergence: 4431.4
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.45978 -0.72728 -0.07774  0.69661  2.90464
##
## Random effects:
##  Groups     Name              Variance  Std.Dev. Corr
##  sent_index (Intercept)       8.289e+00 2.87912
##             gapno-gap         9.193e-02 0.30320   1.00
##             wh_numeric        2.950e-04 0.01717  -1.00 -1.00
##             modifiershort-mod 4.869e-02 0.22066   1.00  1.00 -1.00
##             modifiermed-mod   3.848e-02 0.19616   1.00  1.00 -1.00  1.00
##             modifierlong-mod  1.187e-04 0.01090   1.00  1.00 -1.00  1.00
##  Residual                     5.014e+01 7.08092
##
##
##
##
##
##
##   1.00
##
## Number of obs: 656, groups:  sent_index, 21
##
## Fixed effects:
##                                          Estimate Std. Error        df
## (Intercept)                              11.00250    1.00364  42.19758
## gapno-gap                                -1.09211    1.10784 590.72968
## wh_numeric                               -0.54346    0.78197 619.76614
## modifiershort-mod                        -0.05512    1.10690 587.15871
## modifiermed-mod                          -0.13670    1.10668 593.57227
## modifierlong-mod                         -0.05444    1.10586 619.77059
## gapno-gap:wh_numeric                      0.73419    1.10585 619.85549
## gapno-gap:modifiershort-mod               0.40603    1.56391 619.85549
## gapno-gap:modifiermed-mod                 0.25368    1.56391 619.85549
## gapno-gap:modifierlong-mod                0.13631    1.56391 619.85549
```

```
## wh_numeric:modifiershort-mod             0.34319    1.10585 619.85549
## wh_numeric:modifiermed-mod               0.35869    1.10585 619.85549
## wh_numeric:modifierlong-mod              0.37914    1.10585 619.85549
## gapno-gap:wh_numeric:modifiershort-mod  -0.35776    1.56391 619.85549
## gapno-gap:wh_numeric:modifiermed-mod    -0.42719    1.56391 619.85549
## gapno-gap:wh_numeric:modifierlong-mod   -0.45356    1.56391 619.85549
##                                        t value Pr(>|t|)
## (Intercept)                             10.963 6.32e-14 ***
## gapno-gap                               -0.986    0.325
## wh_numeric                              -0.695    0.487
## modifiershort-mod                       -0.050    0.960
## modifiermed-mod                         -0.124    0.902
## modifierlong-mod                        -0.049    0.961
## gapno-gap:wh_numeric                     0.664    0.507
## gapno-gap:modifiershort-mod              0.260    0.795
## gapno-gap:modifiermed-mod                0.162    0.871
## gapno-gap:modifierlong-mod               0.087    0.931
## wh_numeric:modifiershort-mod             0.310    0.756
## wh_numeric:modifiermed-mod               0.324    0.746
## wh_numeric:modifierlong-mod              0.343    0.732
## gapno-gap:wh_numeric:modifiershort-mod  -0.229    0.819
## gapno-gap:wh_numeric:modifiermed-mod    -0.273    0.785
## gapno-gap:wh_numeric:modifierlong-mod   -0.290    0.772
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
```

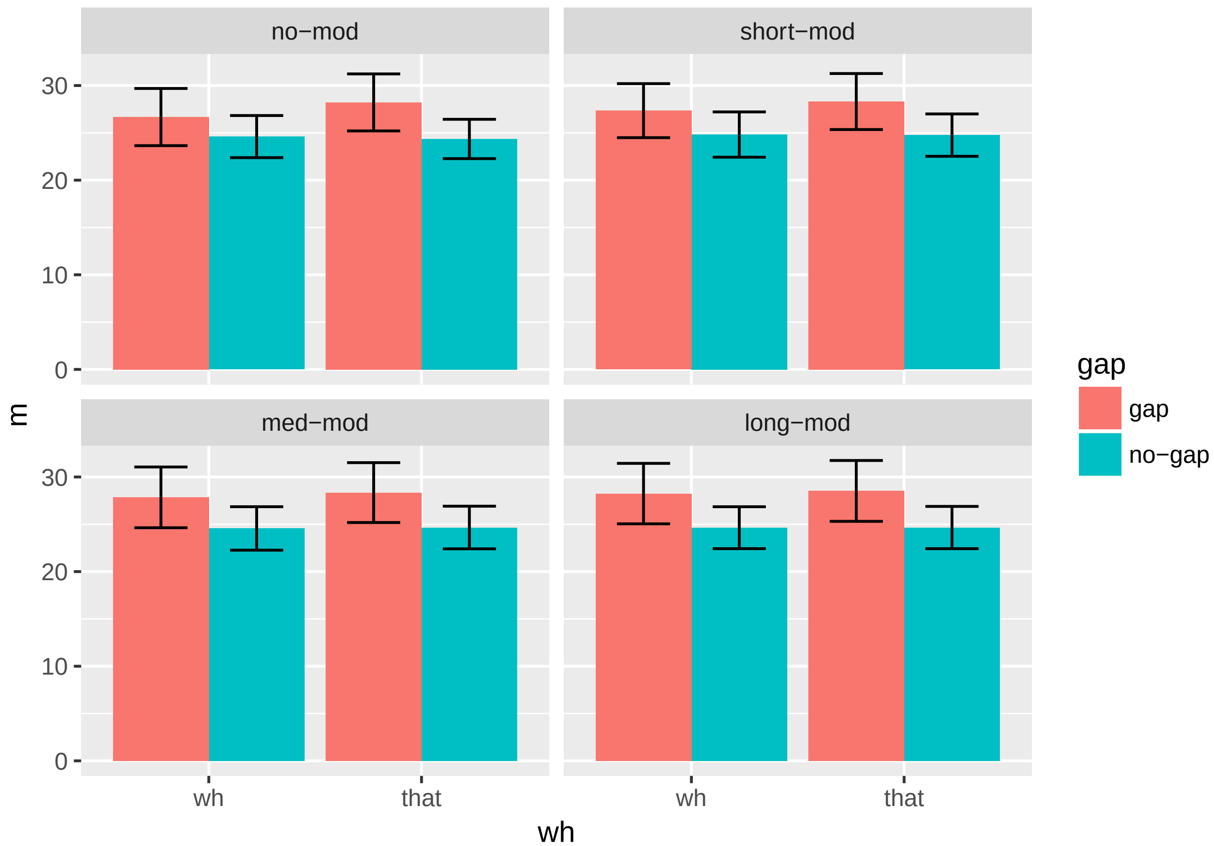
So this says that there is nothing significant in the object position gaps. This goes a little bit against my intuition, based on the graphs plotted above, where the error bars seem well above 0 for the wh/gap interaction.

Gap in indirect object / PP position

```
d2 = d_agg %>%
  filter(model=="google") %>%
  filter(gap_position=="goal") %>%
  filter(region == "temporal_modifier") %>%
  group_by(model, wh, gap, modifier) %>%
    summarise(m=mean(surprisal),
              s=std.error(surprisal),
              upper=m + 1.96*s,
              lower=m - 1.96*s) %>%
    ungroup()

ggplot(d2, aes(x=wh, y=m, ymin=lower, ymax=upper, fill=gap)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~modifier)
```

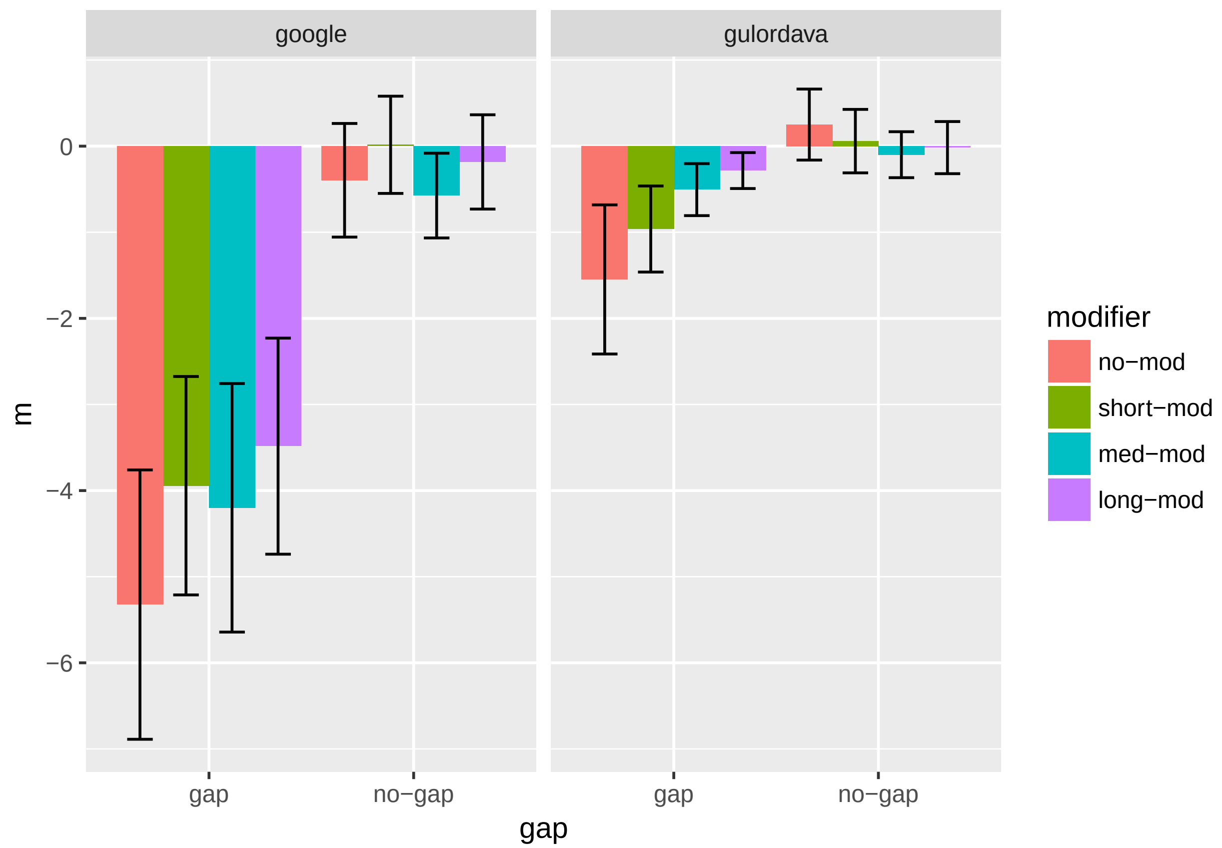



```
d2 = d_agg %>%
  filter(model=="gulordava") %>%
  filter(gap_position=="goal") %>%
  filter(region == "temporal_modifier") %>%
  group_by(model, wh, gap, modifier) %>%
    summarise(m=mean(surprisal),
              s=std.error(surprisal),
              upper=m + 1.96*s,
              lower=m - 1.96*s) %>%
    ungroup()

ggplot(d2, aes(x=wh, y=m, ymin=lower, ymax=upper, fill=gap)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~modifier)
```



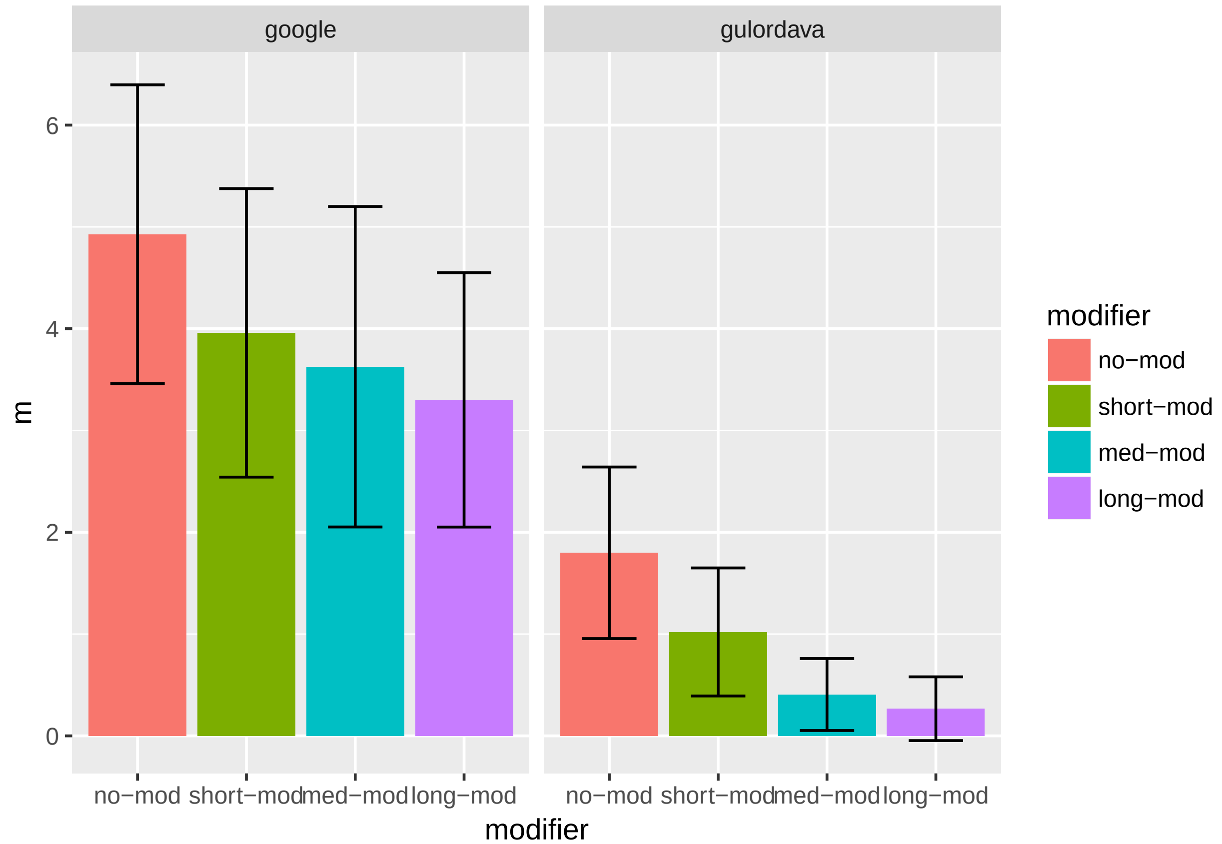
Again we see something very similar to the object gap case. Surprisal between gap and no-gap is about the same in "no modifier" condition when a wh-word is present, but different when it is not.

```
d_wh_effect = d_agg %>%
  filter(region == "temporal_modifier") %>%
  filter(gap_position=="goal") %>%
  select(-wh_numeric) %>%
  spread(wh, surprisal) %>%
  mutate(wh_effect=wh-`that`)

d_wh_effect %>%
  group_by(model, gap, gap_position, modifier) %>%
    summarise(m=mean(wh_effect),
              s=std.error(wh_effect),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
  ggplot(aes(x=gap, y=m, ymin=lower, ymax=upper, fill=modifier)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model)
```



What this shows is that with the pp/goal licensing, when there is a gap, the presence of a wh-word does less to reduce surprisal the further it is away from that gap. It seems, however, that in the no-gap condition, the presence or absence of a wh licensor doesn't really change the surprisal of the network.

```
d_full_interaction = d_agg %>%
  filter(region == "temporal_modifier") %>%
  filter(gap_position=="goal") %>%
  select(-wh_numeric) %>%
  spread(gap, surprisal) %>%
  mutate(gap_effect=`no-gap`-gap) %>%
  select(-unk, -gap, -`no-gap`) %>%
  spread(wh, gap_effect) %>%
  mutate(wh_interaction=wh-`that`)

d_full_interaction %>%
  group_by(model, modifier) %>%
    summarise(m=mean(wh_interaction, na.rm=T),
              s=std.error(wh_interaction, na.rm=T),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
  ggplot(aes(x=modifier, y=m, ymin=lower, ymax=upper, fill=modifier)) +
    geom_bar(stat="identity") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model)
```



Great! This shows very nice decreasing effects of licensing given gap distance in both models, with the google model showing larger licensing effects overall than the gulordava model.

```
m_google = d_agg %>%
  filter(model == "google", region == "temporal_modifier", gap_position=="goal") %>%
  lmer(surprisal ~ gap * wh_numeric * modifier+
                (gap+wh_numeric+modifier|sent_index),
              data=.)
summary(m_google)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: surprisal ~ gap * wh_numeric * modifier + (gap + wh_numeric +
##     modifier | sent_index)
##    Data: .
##
## REML criterion at convergence: 1348.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.99475 -0.62449 -0.03071  0.54998  2.69159
##
## Random effects:
##  Groups     Name              Variance Std.Dev. Corr
##  sent_index (Intercept)       52.9008  7.2733
##             gapno-gap          8.3797  2.8948   -0.58
##             wh_numeric         0.5170  0.7190   -0.16  0.74
```

```
##             modifiershort-mod  0.7977  0.8931   -0.22 -0.10  0.05
##             modifiermed-mod    0.7984  0.8935   -0.33  0.31  0.21  0.84
##             modifierlong-mod   0.9511  0.9752   -0.22  0.31  0.22  0.80
##  Residual                      1.6671  1.2912
##
##
##
##
##
##
##   0.99
##
## Number of obs: 336, groups:  sent_index, 21
##
## Fixed effects:
##                                      Estimate Std. Error       df
## (Intercept)                           20.5182     1.5996  20.2347
## gapno-gap                             -2.7507     0.6917  26.0460
## wh_numeric                            -2.6619     0.2536  84.6797
## modifiershort-mod                      0.9328     0.3426  47.0649
## modifiermed-mod                        1.0379     0.3426  48.8112
## modifierlong-mod                       1.2313     0.3531  44.8396
## gapno-gap:wh_numeric                   2.4639     0.2818 240.0000
## gapno-gap:modifiershort-mod           -0.7778     0.3985 240.0000
## gapno-gap:modifiermed-mod             -0.6933     0.3985 240.0000
## gapno-gap:modifierlong-mod            -0.8268     0.3985 240.0000
## wh_numeric:modifiershort-mod           0.6903     0.2818 240.0000
## wh_numeric:modifiermed-mod             0.5619     0.2818 240.0000
## wh_numeric:modifierlong-mod            0.9200     0.2818 240.0000
## gapno-gap:wh_numeric:modifiershort-mod  -0.4843     0.3985 240.0000
## gapno-gap:wh_numeric:modifiermed-mod  -0.6507     0.3985 240.0000
## gapno-gap:wh_numeric:modifierlong-mod  -0.8134     0.3985 240.0000
##                                      t value Pr(>|t|)
## (Intercept)                           12.827 3.54e-11 ***
## gapno-gap                             -3.977 0.000495 ***
## wh_numeric                           -10.497  < 2e-16 ***
## modifiershort-mod                      2.723 0.009050 **
## modifiermed-mod                        3.029 0.003912 **
## modifierlong-mod                       3.487 0.001105 **
## gapno-gap:wh_numeric                   8.745 3.98e-16 ***
## gapno-gap:modifiershort-mod           -1.952 0.052090 .
## gapno-gap:modifiermed-mod             -1.740 0.083139 .
## gapno-gap:modifierlong-mod            -2.075 0.039059 *
## wh_numeric:modifiershort-mod           2.450 0.015002 *
## wh_numeric:modifiermed-mod             1.994 0.047253 *
## wh_numeric:modifierlong-mod            3.265 0.001253 **
## gapno-gap:wh_numeric:modifiershort-mod  -1.215 0.225430
## gapno-gap:wh_numeric:modifiermed-mod  -1.633 0.103767
## gapno-gap:wh_numeric:modifierlong-mod  -2.041 0.042300 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 16 > 12.
```

```
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
```

```
m_gul = d_agg %>%
  filter(model == "gulordava", region == "temporal_modifier", gap_position=="goal") %>%
  lmer(surprisal ~ gap * wh_numeric * modifier+
                  (gap+wh_numeric+modifier|sent_index),
                data=.)
summary(m_gul)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: surprisal ~ gap * wh_numeric * modifier + (gap + wh_numeric +
##     modifier | sent_index)
##    Data: .
##
## REML criterion at convergence: 1071.7
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.4165 -0.4290 -0.0254  0.4559  2.7834
##
## Random effects:
##  Groups     Name              Variance Std.Dev. Corr
##  sent_index (Intercept)       47.7036  6.9068
##             gapno-gap         21.9975  4.6902   -0.67
##             wh_numeric         0.0488  0.2209    0.10  0.26
##             modifiershort-mod  0.4460  0.6678    0.11 -0.29 -0.52
##             modifiermed-mod    1.9334  1.3905    0.20 -0.23 -0.15  0.76
##             modifierlong-mod   2.2075  1.4858    0.18 -0.34 -0.34  0.82
##  Residual                      0.5080  0.7128
##
##
##
##
##
##
##   0.95
##
## Number of obs: 336, groups:  sent_index, 21
##
## Fixed effects:
##                                         Estimate Std. Error       df
## (Intercept)                              27.4438     1.5112  20.0797
## gapno-gap                                -2.9653     1.0352  20.6942
## wh_numeric                               -0.7738     0.1201 160.4695
## modifiershort-mod                         0.3830     0.2131  36.6661
## modifiermed-mod                           0.6537     0.3410  24.8812
## modifierlong-mod                          0.9416     0.3596  24.3149
## gapno-gap:wh_numeric                      0.8992     0.1555 200.0000
## gapno-gap:modifiershort-mod              -0.0706     0.2200 200.0000
## gapno-gap:modifiermed-mod                -0.5222     0.2200 200.0000
## gapno-gap:modifierlong-mod               -0.7726     0.2200 200.0000
## wh_numeric:modifiershort-mod              0.2928     0.1555 200.0000
## wh_numeric:modifiermed-mod                0.5213     0.1555 200.0000
```

```
## wh_numeric:modifierlong-mod               0.6320     0.1555 200.0000
## gapno-gap:wh_numeric:modifiershort-mod  -0.3888     0.2200 200.0000
## gapno-gap:wh_numeric:modifiermed-mod    -0.6962     0.2200 200.0000
## gapno-gap:wh_numeric:modifierlong-mod   -0.7658     0.2200 200.0000
##                                         t value Pr(>|t|)
## (Intercept)                              18.160 6.25e-14 ***
## gapno-gap                                -2.864 0.009371 **
## wh_numeric                               -6.444 1.30e-09 ***
## modifiershort-mod                         1.797 0.080590 .
## modifiermed-mod                           1.917 0.066761 .
## modifierlong-mod                          2.618 0.014976 *
## gapno-gap:wh_numeric                      5.781 2.82e-08 ***
## gapno-gap:modifiershort-mod              -0.321 0.748558
## gapno-gap:modifiermed-mod                -2.374 0.018542 *
## gapno-gap:modifierlong-mod               -3.512 0.000549 ***
## wh_numeric:modifiershort-mod              1.882 0.061251 .
## wh_numeric:modifiermed-mod                3.351 0.000961 ***
## wh_numeric:modifierlong-mod               4.063 6.95e-05 ***
## gapno-gap:wh_numeric:modifiershort-mod   -1.768 0.078660 .
## gapno-gap:wh_numeric:modifiermed-mod     -3.165 0.001792 **
## gapno-gap:wh_numeric:modifierlong-mod    -3.482 0.000612 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
```

Unlike in the case of gaps in object position, when the gap is in pp/goal position we start to see some significant effects. Particularly (as we predicted) there is an interaction between gaps and wh-words, such that when you have no gap, but you do have a wh-licensor there is a significant increase in surprisal by about 1 bit of information. Also important, we see an interaction between gaps/wh-licensors and distance. In the google model the interaction is only significiant when the modifier is long, which is what we predicted. In the gulordava model we see a significant interaction in both the long and medium filler cases, but the significance is greater and the effect size is bigger in the long case. (Although the difference in effect size is only about 0.1 bits of surprisal.)

Now instead of looking at surprisal directly post-gap, we move on to the entire embedded region.

Surprisal in the entire embedded region

```
remove_na = function(x) {
  x[!is.na(x)]
}

d = read_csv("tests/combined_results.csv") %>%
  select(-1, -2) %>%
  mutate(unk=unk == "True") %>%
  mutate(region=if_else(region=="prefix" | region=="obj wh" | region=="goal wh" | region=="that", "prefi
  mutate(region=if_else(region=="short modifier" | region=="medium modifier" | region=="long modifier"
  separate(condition, sep="_", into=c("wh", "gap", "gap_position", "modifier"))
```

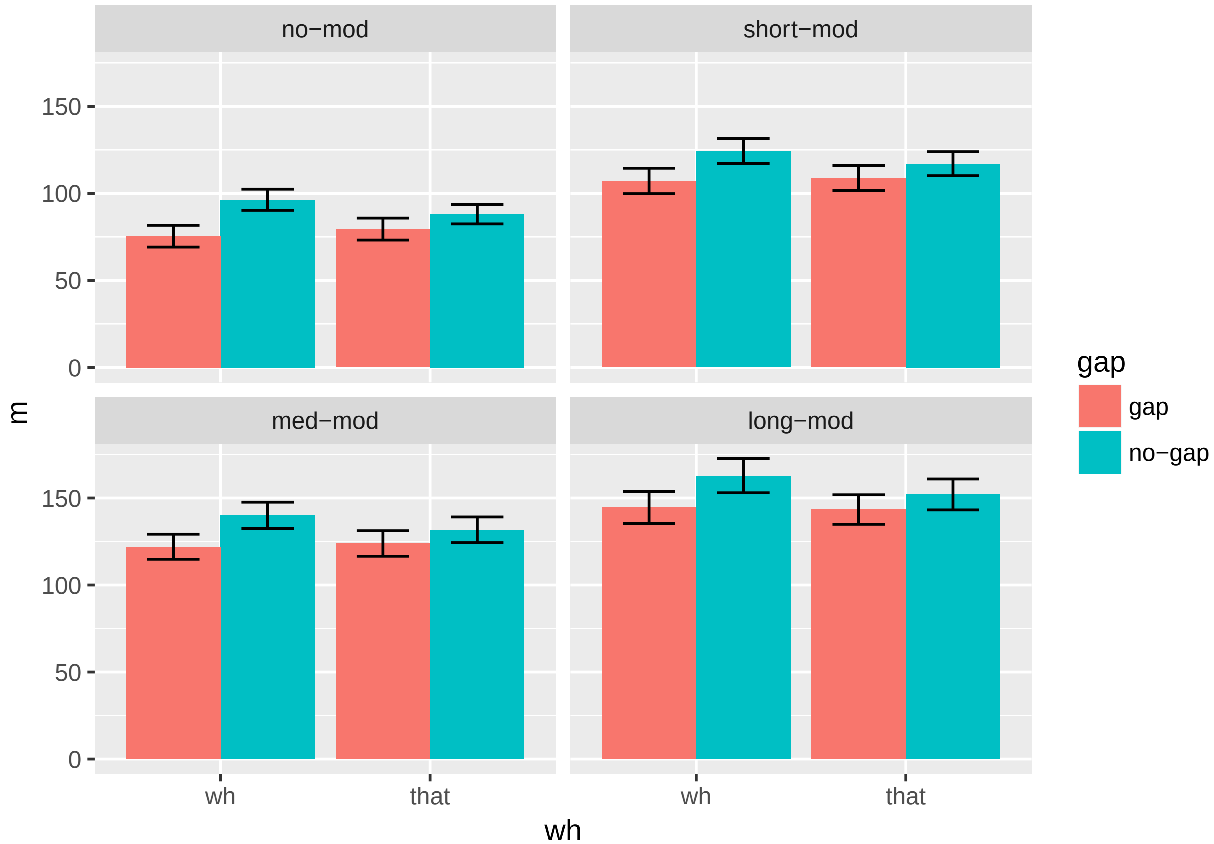
```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_integer(),
##   `Unnamed: 0` = col_integer(),
##   sent_index = col_integer(),
##   word_index = col_integer(),
##   word = col_character(),
##   region = col_character(),
##   condition = col_character(),
##   model_word = col_character(),
##   surprisal = col_double(),
##   model = col_character(),
##   unk = col_character()
## )
```

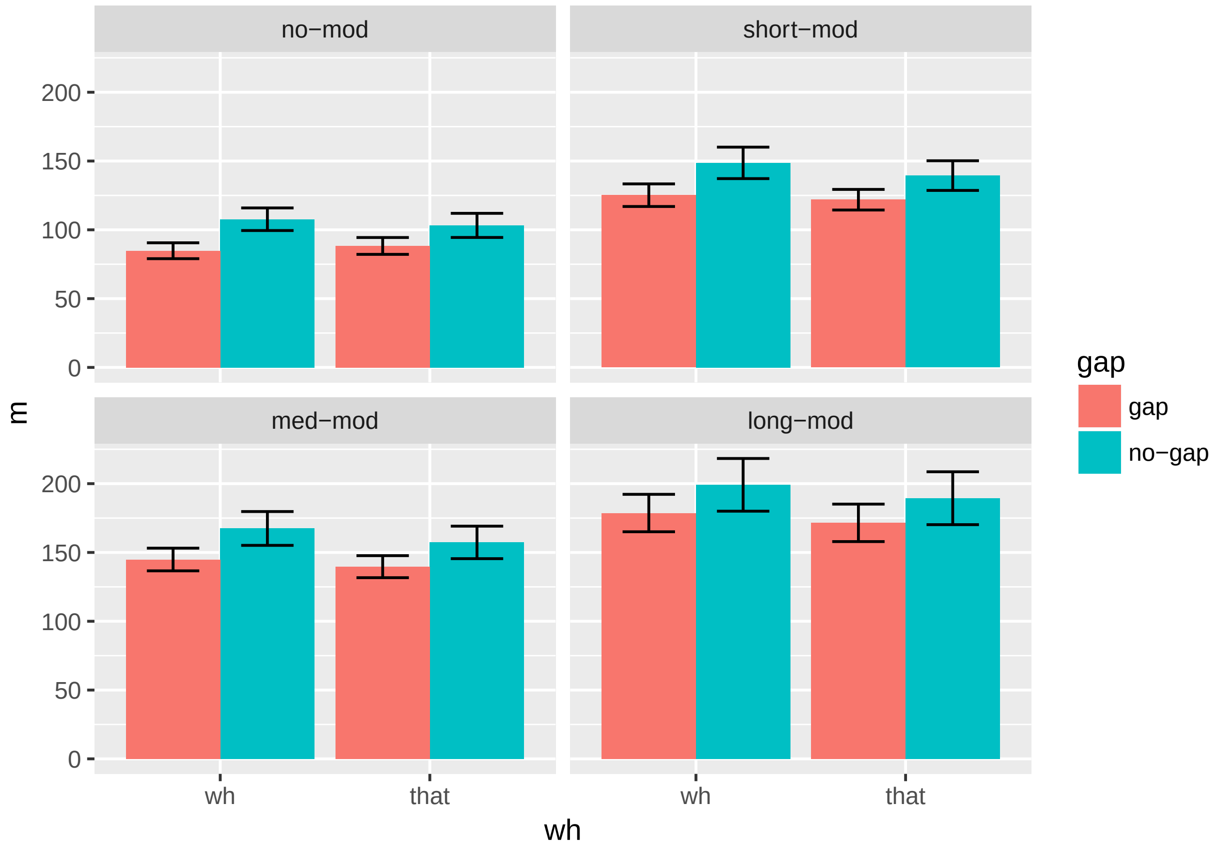
```
d_agg = d %>%
  group_by(model, region, sent_index, wh, gap, gap_position, modifier) %>%
    summarise(surprisal=sum(surprisal),
              unk=any(unk)) %>%
    ungroup() %>%
  filter(!unk) %>%
  mutate(wh_numeric=if_else(wh == "wh", 1, -1),
         wh=factor(wh, levels=c("wh", "that")),
         gap=factor(gap, levels=c("gap", "no-gap")),
         gap_position=factor(gap_position, levels=c("obj", "goal")),
         modifier=factor(modifier, levels=c("no-mod", "short-mod", "med-mod", "long-mod")))
```

Okay, let's do a quick visualization to see what's going on here.

```
d2 = d_agg %>%
  filter(model=="google") %>%
  filter(region=="embed") %>%
  filter(gap_position=="obj") %>%
  group_by(model, wh, gap, modifier) %>%
    summarise(m=mean(surprisal),
              s=std.error(surprisal),
              upper=m + 1.96*s,
              lower=m - 1.96*s) %>%
    ungroup()

ggplot(d2, aes(x=wh, y=m, ymin=lower, ymax=upper, fill=gap)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~modifier)
```

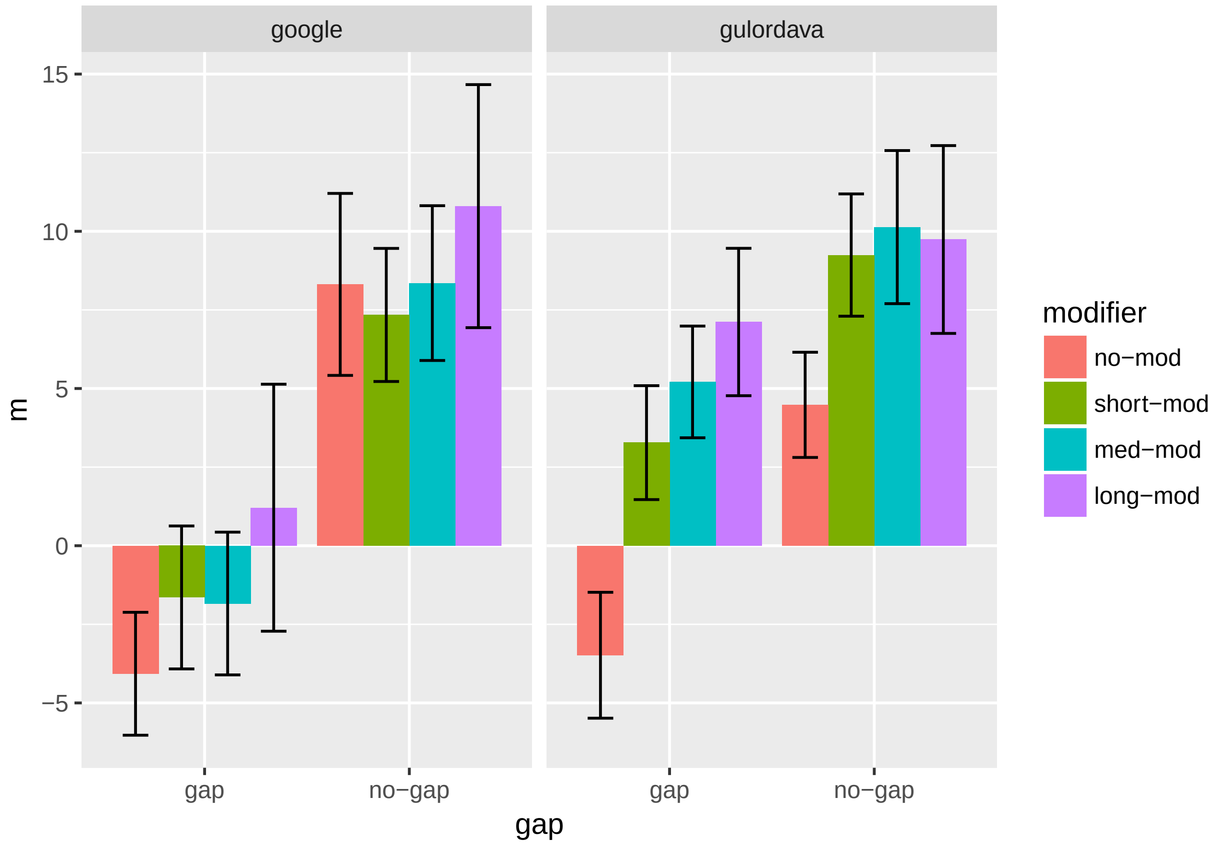



```
d2 = d_agg %>%
  filter(model=="gulordava") %>%
    filter(region=="embed") %>%
  filter(gap_position=="obj") %>%
  group_by(model, wh, gap, modifier) %>%
    summarise(m=mean(surprisal),
              s=std.error(surprisal),
              upper=m + 1.96*s,
              lower=m - 1.96*s) %>%
    ungroup()

ggplot(d2, aes(x=wh, y=m, ymin=lower, ymax=upper, fill=gap)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~modifier)
```

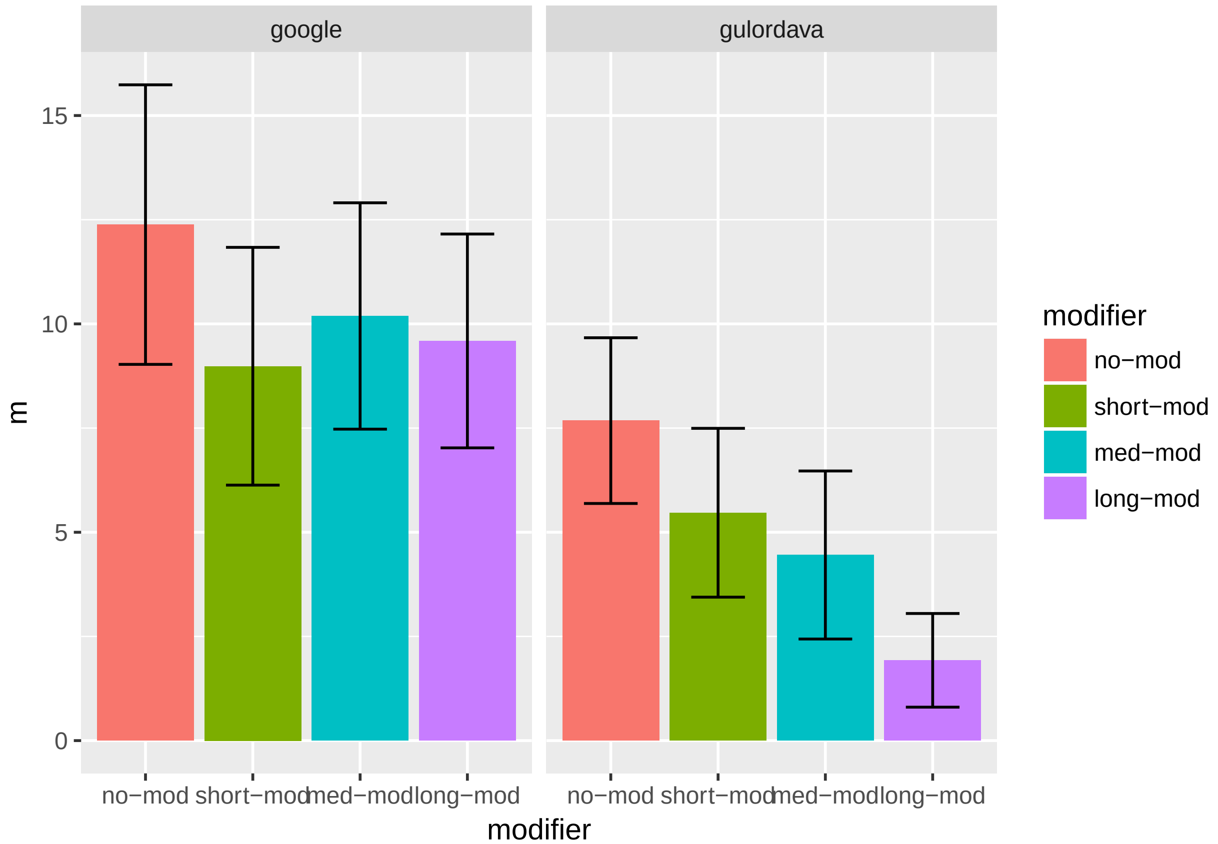


From first glance it does not look like any of the effects we observed in the above sections have translated into effects for the entire embedded region. In all conditions it looks like the "no-gap" surprisal is higher.

```
d_wh_effect = d_agg %>%
  filter(gap_position=="obj") %>%
    filter(region=="embed") %>%
  select(-wh_numeric) %>%
  spread(wh, surprisal) %>%
  mutate(wh_effect=wh-`that`)

d_wh_effect %>%
  group_by(model, gap, gap_position, modifier) %>%
    summarise(m=mean(wh_effect),
              s=std.error(wh_effect),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
  ggplot(aes(x=gap, y=m, ymin=lower, ymax=upper, fill=modifier)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model)
```




```
d_full_interaction = d_agg %>%
  filter(region=="embed") %>%
  filter(gap_position=="obj") %>%
  select(-wh_numeric) %>%
  spread(gap, surprisal) %>%
  mutate(gap_effect=`no-gap`-gap) %>%
  select(-unk, -gap, -`no-gap`) %>%
  spread(wh, gap_effect) %>%
  mutate(wh_interaction=wh-`that`)

d_full_interaction %>%
  group_by(model, modifier) %>%
    summarise(m=mean(wh_interaction, na.rm=T),
              s=std.error(wh_interaction, na.rm=T),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
  ggplot(aes(x=modifier, y=m, ymin=lower, ymax=upper, fill=modifier)) +
    geom_bar(stat="identity") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model)
```



Statistics:

```
m_google = d_agg %>%
  filter(model == "google", region == "embed", gap_position=="obj") %>%
  lmer(surprisal ~ gap * wh_numeric * modifier+
                (gap+wh_numeric+modifier|sent_index),
              data=.)
summary(m_google)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: surprisal ~ gap * wh_numeric * modifier + (gap + wh_numeric +
##     modifier | sent_index)
##    Data: .
##
## REML criterion at convergence: 2067.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.2673 -0.4617  0.0103  0.4661  2.7563
##
## Random effects:
##  Groups     Name              Variance Std.Dev. Corr
##  sent_index (Intercept)       201.024  14.178
##             gapno-gap          82.773   9.098   -0.41
##             wh_numeric          5.200   2.280    0.23 -0.12
##             modifiershort-mod  33.898   5.822    0.31 -0.05 -0.30
```

```
##             modifiermed-mod    58.872   7.673    0.11  0.15 -0.39  0.79
##             modifierlong-mod  181.267  13.464    0.04  0.34  0.18  0.43
##  Residual                      9.685   3.112
##
##
##
##
##
##
##   0.30
##
## Number of obs: 336, groups:  sent_index, 21
##
## Fixed effects:
##                                        Estimate Std. Error       df
## (Intercept)                             77.4414     3.1310  20.3574
## gapno-gap                               14.7521     2.0983  23.5389
## wh_numeric                              -2.0359     0.6915  56.8221
## modifiershort-mod                       30.5045     1.4406  25.2731
## modifiermed-mod                         45.5044     1.8068  23.1424
## modifierlong-mod                        66.5344     3.0155  21.0529
## gapno-gap:wh_numeric                     6.1914     0.6791 200.0000
## gapno-gap:modifiershort-mod             -2.0264     0.9604 200.0000
## gapno-gap:modifiermed-mod               -1.8209     0.9604 200.0000
## gapno-gap:modifierlong-mod              -1.2949     0.9604 200.0000
## wh_numeric:modifiershort-mod             1.2135     0.6791 200.0000
## wh_numeric:modifiermed-mod               1.1170     0.6791 200.0000
## wh_numeric:modifierlong-mod              2.6403     0.6791 200.0000
## gapno-gap:wh_numeric:modifiershort-mod  -1.6993     0.9604 200.0000
## gapno-gap:wh_numeric:modifiermed-mod    -1.0963     0.9604 200.0000
## gapno-gap:wh_numeric:modifierlong-mod   -1.3960     0.9604 200.0000
##                                        t value Pr(>|t|)
## (Intercept)                             24.734  < 2e-16 ***
## gapno-gap                                7.031 3.20e-07 ***
## wh_numeric                              -2.944 0.004687 **
## modifiershort-mod                       21.174  < 2e-16 ***
## modifiermed-mod                         25.185  < 2e-16 ***
## modifierlong-mod                        22.064 4.91e-16 ***
## gapno-gap:wh_numeric                     9.117  < 2e-16 ***
## gapno-gap:modifiershort-mod             -2.110 0.036115 *
## gapno-gap:modifiermed-mod               -1.896 0.059412 .
## gapno-gap:modifierlong-mod              -1.348 0.179114
## wh_numeric:modifiershort-mod             1.787 0.075483 .
## wh_numeric:modifiermed-mod               1.645 0.101587
## wh_numeric:modifierlong-mod              3.888 0.000138 ***
## gapno-gap:wh_numeric:modifiershort-mod  -1.769 0.078363 .
## gapno-gap:wh_numeric:modifiermed-mod    -1.142 0.255025
## gapno-gap:wh_numeric:modifierlong-mod   -1.453 0.147659
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
```

```
##   vcov(x)     if you need it
```

```r
m_gul = d_agg %>%
  filter(model == "gulordava", region == "embed", gap_position=="obj") %>%
  lmer(surprisal ~ gap * wh_numeric * modifier+
                  (gap+wh_numeric+modifier|sent_index),
                data=.)
summary(m_gul)
```

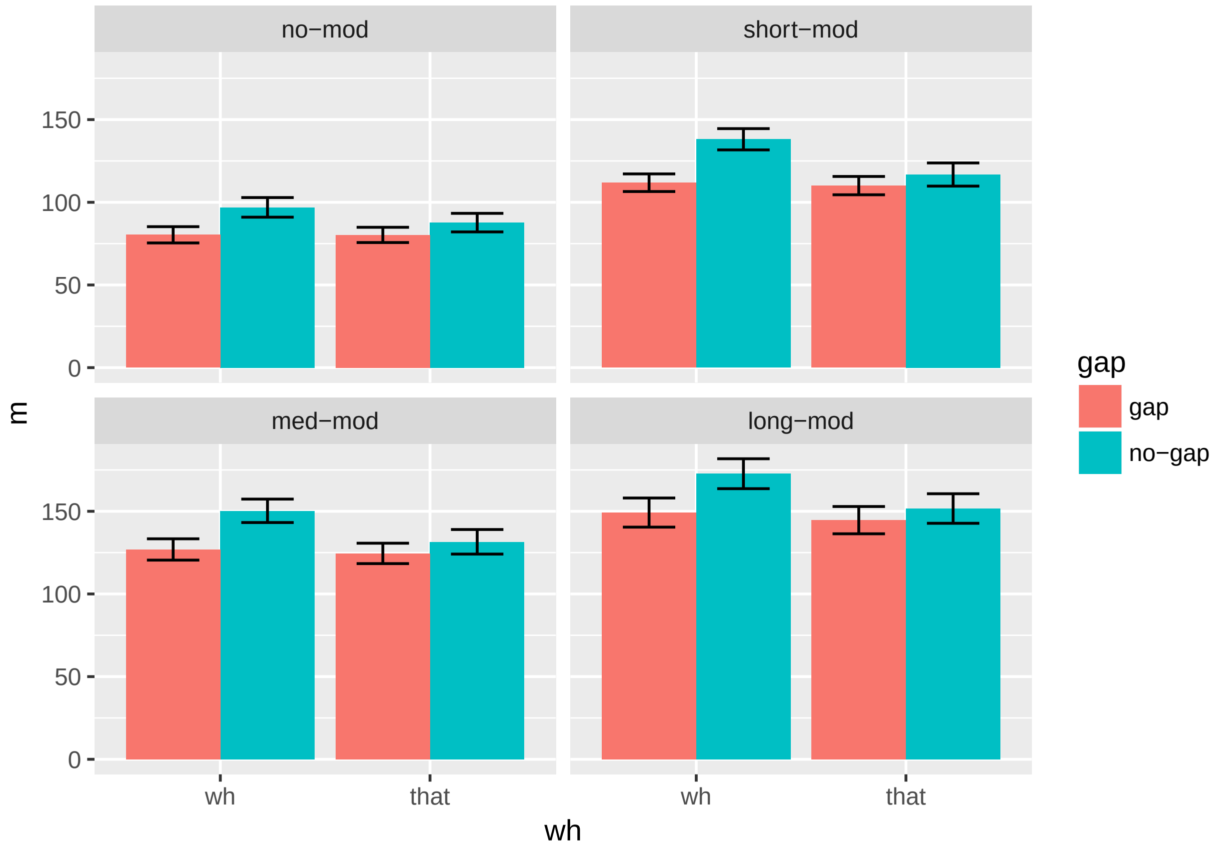
```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: surprisal ~ gap * wh_numeric * modifier + (gap + wh_numeric +
##     modifier | sent_index)
##    Data: .
##
## REML criterion at convergence: 1530.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -2.44473 -0.46353 -0.00393  0.50709  2.22062
##
## Random effects:
##  Groups     Name              Variance Std.Dev. Corr
##  sent_index (Intercept)       161.170  12.695
##             gapno-gap         101.632  10.081    0.26
##             wh_numeric          2.216   1.488   -0.06 -0.41
##             modifiershort-mod  45.049   6.712    0.45  0.18  0.22
##             modifiermed-mod    69.660   8.346    0.33  0.38 -0.02  0.76
##             modifierlong-mod  384.899  19.619    0.31  0.24  0.39  0.50
##  Residual                       5.267   2.295
##
##
##
##
##
##
##   0.39
##
## Number of obs: 264, groups:  sent_index, 19
##
## Fixed effects:
##                                      Estimate Std. Error       df
## (Intercept)                           86.5424     2.9362  18.1704
## gapno-gap                             19.7702     2.4620  18.4118
## wh_numeric                            -1.7415     0.5052  55.0511
## modifiershort-mod                     37.5718     1.6622  19.6078
## modifiermed-mod                       56.3671     2.0291  18.8365
## modifierlong-mod                      85.6772     4.8903  16.4433
## gapno-gap:wh_numeric                   3.8254     0.5445 148.5063
## gapno-gap:modifiershort-mod            1.3571     0.8059 150.9270
## gapno-gap:modifiermed-mod              1.2234     0.8083 149.5933
## gapno-gap:modifierlong-mod             1.1307     0.8689 149.1014
## wh_numeric:modifiershort-mod           3.2895     0.5359 148.6517
## wh_numeric:modifiermed-mod             4.2547     0.5359 148.6517
## wh_numeric:modifierlong-mod            5.0025     0.5660 149.5155
```

```
## gapno-gap:wh_numeric:modifiershort-mod  -1.0200     0.7774 147.4544
## gapno-gap:wh_numeric:modifiermed-mod    -1.5407     0.7774 147.4544
## gapno-gap:wh_numeric:modifierlong-mod   -2.7632     0.8197 147.4954
##                                        t value Pr(>|t|)
## (Intercept)                             29.474  < 2e-16 ***
## gapno-gap                                8.030 1.98e-07 ***
## wh_numeric                              -3.447 0.001092 **
## modifiershort-mod                       22.604 1.64e-15 ***
## modifiermed-mod                         27.779  < 2e-16 ***
## modifierlong-mod                        17.520 4.58e-12 ***
## gapno-gap:wh_numeric                     7.026 7.14e-11 ***
## gapno-gap:modifiershort-mod              1.684 0.094258 .
## gapno-gap:modifiermed-mod                1.514 0.132220
## gapno-gap:modifierlong-mod               1.301 0.195189
## wh_numeric:modifiershort-mod             6.139 7.21e-09 ***
## wh_numeric:modifiermed-mod               7.940 4.57e-13 ***
## wh_numeric:modifierlong-mod              8.839 2.47e-15 ***
## gapno-gap:wh_numeric:modifiershort-mod  -1.312 0.191542
## gapno-gap:wh_numeric:modifiermed-mod    -1.982 0.049365 *
## gapno-gap:wh_numeric:modifierlong-mod   -3.371 0.000957 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
```

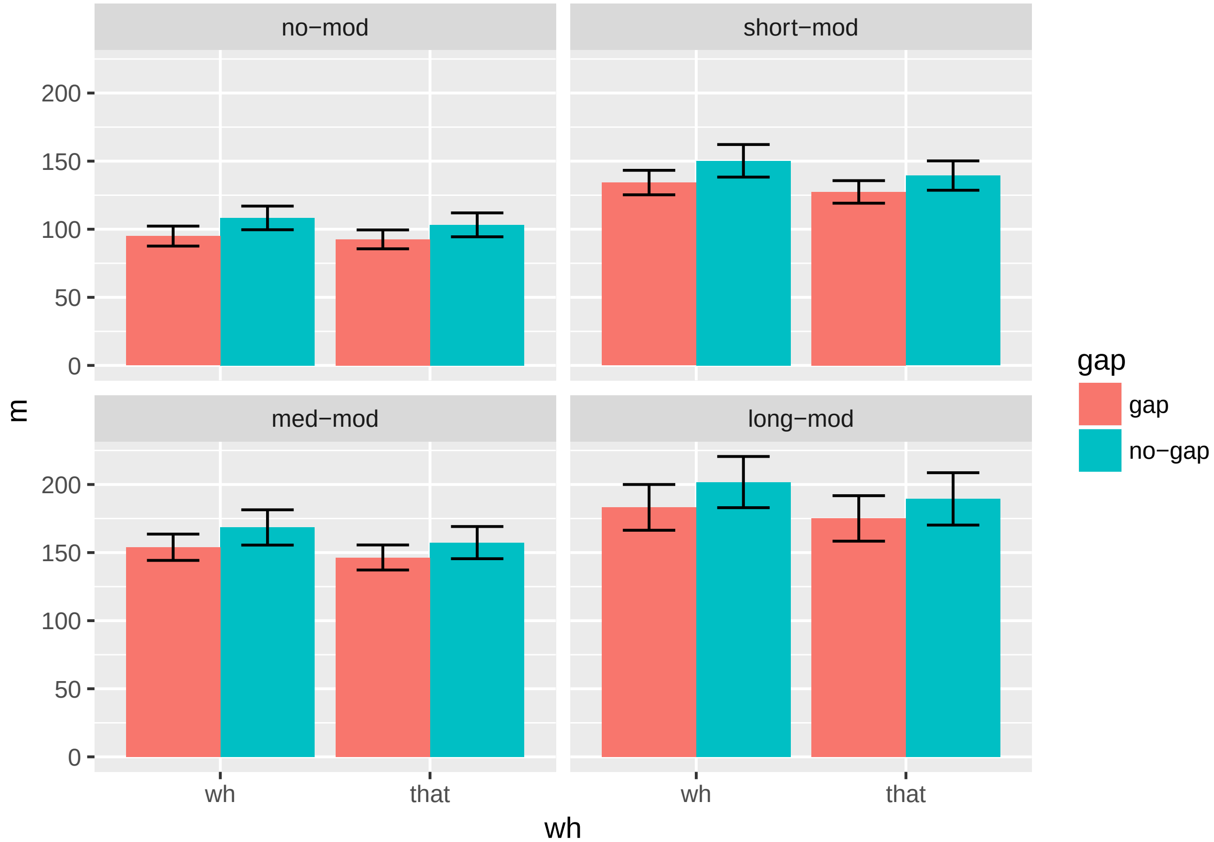
So it appears that the interaction of the filler/gap dependency does actually have a significant effect on surprisal and, unlike when we measured just the post-gap material, it also translates into a significant three-way interaction between gap,wh-word and length. In this case, the long modifier has a more significant and greater effect (although we do not test here whether the difference is effect is significant).

Now for the pp/goal position gap

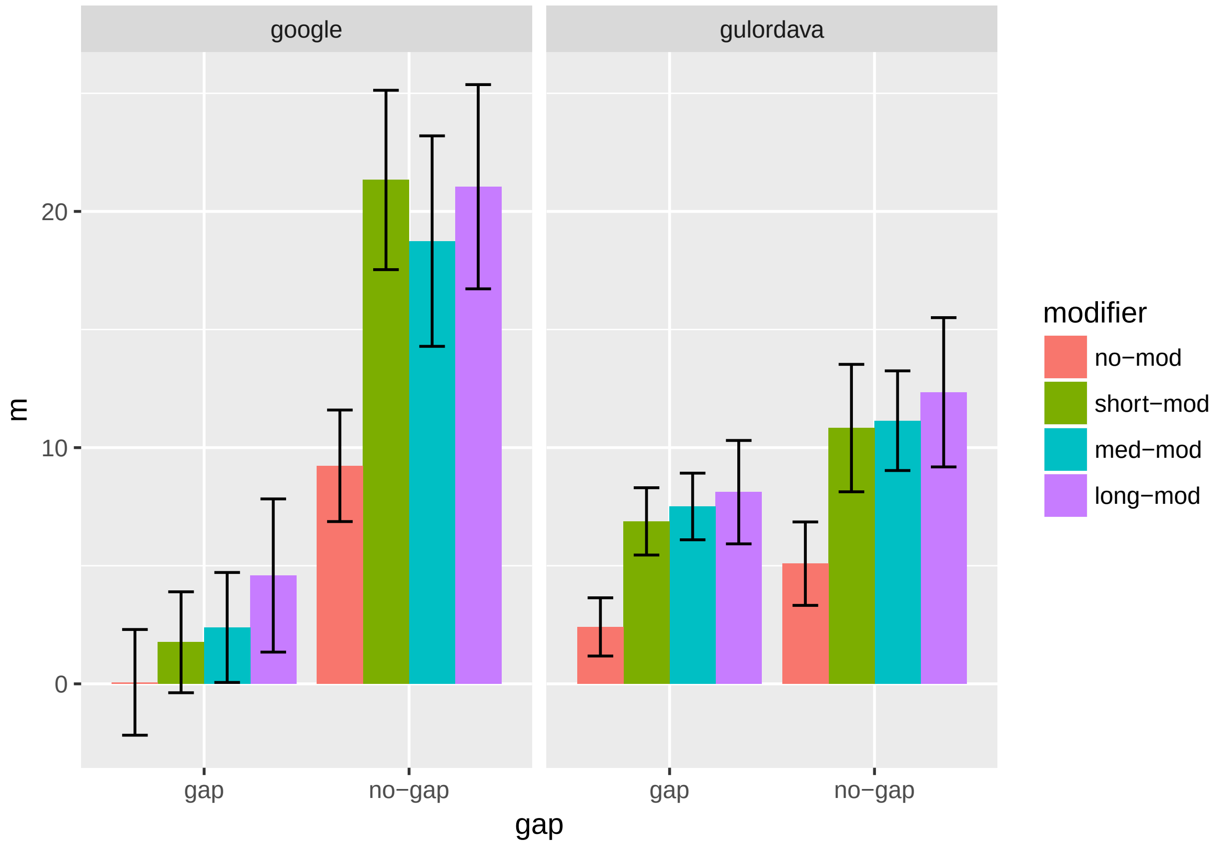
```
d2 = d_agg %>%
  filter(model=="google") %>%
  filter(region=="embed") %>%
  filter(gap_position=="goal") %>%
  group_by(model, wh, gap, modifier) %>%
    summarise(m=mean(surprisal),
              s=std.error(surprisal),
              upper=m + 1.96*s,
              lower=m - 1.96*s) %>%
    ungroup()

ggplot(d2, aes(x=wh, y=m, ymin=lower, ymax=upper, fill=gap)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~modifier)
```




```
d2 = d_agg %>%
  filter(model=="gulordava") %>%
    filter(region=="embed") %>%
  filter(gap_position=="goal") %>%
  group_by(model, wh, gap, modifier) %>%
    summarise(m=mean(surprisal),
              s=std.error(surprisal),
              upper=m + 1.96*s,
              lower=m - 1.96*s) %>%
    ungroup()

ggplot(d2, aes(x=wh, y=m, ymin=lower, ymax=upper, fill=gap)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~modifier)
```

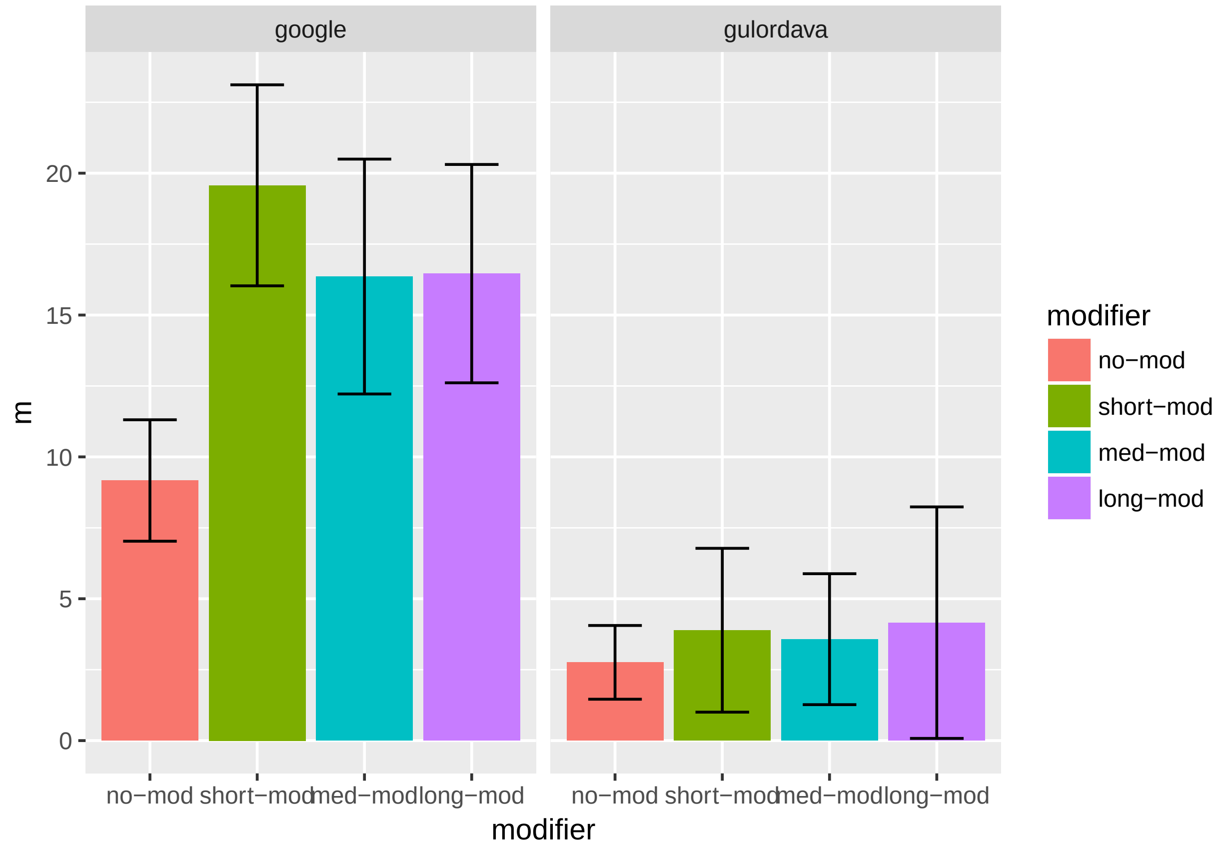



```r
d_wh_effect = d_agg %>%
  filter(gap_position=="goal") %>%
    filter(region=="embed") %>%
  select(-wh_numeric) %>%
  spread(wh, surprisal) %>%
  mutate(wh_effect=wh-`that`)

d_wh_effect %>%
  group_by(model, gap, gap_position, modifier) %>%
    summarise(m=mean(wh_effect),
              s=std.error(wh_effect),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
  ggplot(aes(x=gap, y=m, ymin=lower, ymax=upper, fill=modifier)) +
    geom_bar(stat="identity", position="dodge") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model)
```



In every instance you get more surprisal for the presence of a wh-word, although (at least for the google model you get much more surprisal for a no-gap than a gap). Maybe this just means that, in general, it's more difficult for the network to process wh-headed relative clauses.

```
d_full_interaction = d_agg %>%
  filter(region=="embed") %>%
  filter(gap_position=="goal") %>%
  select(-wh_numeric) %>%
  spread(gap, surprisal) %>%
  mutate(gap_effect=`no-gap`-gap) %>%
  select(-unk, -gap, -`no-gap`) %>%
  spread(wh, gap_effect) %>%
  mutate(wh_interaction=wh-`that`)

d_full_interaction %>%
  group_by(model, modifier) %>%
    summarise(m=mean(wh_interaction, na.rm=T),
              s=std.error(wh_interaction, na.rm=T),
              upper=m+1.96*s,
              lower=m-1.96*s) %>%
    ungroup() %>%
  ggplot(aes(x=modifier, y=m, ymin=lower, ymax=upper, fill=modifier)) +
    geom_bar(stat="identity") +
    geom_errorbar(color="black", width=.5, position=position_dodge(width=.9)) +
    facet_wrap(~model)
```



Statistics:

```
m_google = d_agg %>%
  filter(model == "google", region == "embed", gap_position=="goal") %>%
  lmer(surprisal ~ gap * wh_numeric * modifier+
                (gap+wh_numeric+modifier|sent_index),
              data=.)
summary(m_google)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: surprisal ~ gap * wh_numeric * modifier + (gap + wh_numeric +
##     modifier | sent_index)
##    Data: .
##
## REML criterion at convergence: 2111.2
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.9234 -0.4603 -0.0189  0.4145  3.3056
##
## Random effects:
##  Groups     Name               Variance Std.Dev. Corr
##  sent_index (Intercept)        114.405  10.696
##             gapno-gap           33.261   5.767    0.20
##             wh_numeric           7.308   2.703    0.16 -0.13
##             modifiershort-mod  30.685   5.539    0.05  0.19 -0.45
```

```
##             modifiermed-mod    50.547   7.110    0.24  0.01 -0.39  0.72
##             modifierlong-mod  202.602  14.234    0.13  0.07 -0.15  0.55
##  Residual                      12.676   3.560
##
##
##
##
##
##
##   0.34
##
## Number of obs: 336, groups:  sent_index, 21
##
## Fixed effects:
##                                         Estimate Std. Error        df
## (Intercept)                             80.30265    2.39785  20.81042
## gapno-gap                               12.01497    1.47902  31.58547
## wh_numeric                               0.03143    0.80611  54.21858
## modifiershort-mod                       30.65354    1.43694  27.35358
## modifiermed-mod                         45.41574    1.73511  24.67435
## modifierlong-mod                        66.62831    3.20177  21.22981
## gapno-gap:wh_numeric                     4.58401    0.77693 199.99998
## gapno-gap:modifiershort-mod              4.48414    1.09875 199.99998
## gapno-gap:modifiermed-mod                3.17685    1.09875 199.99998
## gapno-gap:modifierlong-mod               3.25738    1.09875 199.99998
## wh_numeric:modifiershort-mod             0.84839    0.77693 199.99998
## wh_numeric:modifiermed-mod               1.16152    0.77693 199.99998
## wh_numeric:modifierlong-mod              2.26145    0.77693 199.99998
## gapno-gap:wh_numeric:modifiershort-mod   5.20196    1.09875 199.99998
## gapno-gap:wh_numeric:modifiermed-mod     3.59445    1.09875 199.99998
## gapno-gap:wh_numeric:modifierlong-mod    3.64548    1.09875 199.99998
##                                        t value Pr(>|t|)
## (Intercept)                             33.489  < 2e-16 ***
## gapno-gap                                8.124 3.10e-09 ***
## wh_numeric                               0.039  0.96904
## modifiershort-mod                       21.332  < 2e-16 ***
## modifiermed-mod                         26.175  < 2e-16 ***
## modifierlong-mod                        20.810 1.32e-15 ***
## gapno-gap:wh_numeric                     5.900 1.53e-08 ***
## gapno-gap:modifiershort-mod              4.081 6.47e-05 ***
## gapno-gap:modifiermed-mod                2.891  0.00426 **
## gapno-gap:modifierlong-mod               2.965  0.00340 **
## wh_numeric:modifiershort-mod             1.092  0.27616
## wh_numeric:modifiermed-mod               1.495  0.13649
## wh_numeric:modifierlong-mod              2.911  0.00401 **
## gapno-gap:wh_numeric:modifiershort-mod   4.734 4.15e-06 ***
## gapno-gap:wh_numeric:modifiermed-mod     3.271  0.00126 **
## gapno-gap:wh_numeric:modifierlong-mod    3.318  0.00108 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
```

```
##   vcov(x)     if you need it
```

```
m_gul = d_agg %>%
  filter(model == "gulordava", region == "embed", gap_position=="goal") %>%
  lmer(surprisal ~ gap * wh_numeric * modifier+
                  (gap+wh_numeric+modifier|sent_index),
                data=.)
```

```
## Warning in optwrap(optimizer, devfun, getStart(start, rho$lower, rho$pp), :
## convergence code 1 from bobyqa: bobyqa -- maximum number of function
## evaluations exceeded
```

```
summary(m_gul)
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: surprisal ~ gap * wh_numeric * modifier + (gap + wh_numeric +
##     modifier | sent_index)
##    Data: .
##
## REML criterion at convergence: 1488.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.1433 -0.4372  0.0392  0.4128  2.4137
##
## Random effects:
##  Groups     Name              Variance Std.Dev. Corr
##  sent_index (Intercept)       226.0399 15.0346
##             gapno-gap          45.2184  6.7245  0.29
##             wh_numeric          0.6427  0.8017  0.34 0.44
##             modifiershort-mod  59.0384  7.6836  0.15 0.79 0.28
##             modifiermed-mod    79.3652  8.9087  0.31 0.76 0.35 0.82
##             modifierlong-mod  479.5605 21.8989  0.29 0.43 0.52 0.58 0.43
##  Residual                       6.6594  2.5806
## Number of obs: 256, groups:  sent_index, 18
##
## Fixed effects:
##                                         Estimate Std. Error       df
## (Intercept)                              93.7611     3.5697  17.1732
## gapno-gap                                11.4566     1.7154  19.9059
## wh_numeric                                1.2050     0.4698 122.6174
## modifiershort-mod                        37.9264     1.9338  18.7877
## modifiermed-mod                          57.3121     2.2155  18.5138
## modifierlong-mod                         86.5839     5.6213  16.4378
## gapno-gap:wh_numeric                      1.3164     0.6179 147.0829
## gapno-gap:modifiershort-mod               1.3354     0.9082 148.5440
## gapno-gap:modifiermed-mod                 0.3872     0.9086 148.2941
## gapno-gap:modifierlong-mod                0.1753     0.9752 148.8536
## wh_numeric:modifiershort-mod              2.2264     0.6181 147.5058
## wh_numeric:modifiermed-mod                2.5420     0.6181 147.5058
## wh_numeric:modifierlong-mod               2.8038     0.6537 149.4990
## gapno-gap:wh_numeric:modifiershort-mod    0.6342     0.8859 146.1664
## gapno-gap:wh_numeric:modifiermed-mod      0.4747     0.8859 146.1664
## gapno-gap:wh_numeric:modifierlong-mod     0.7646     0.9354 146.2042
```

```
##                                          t value Pr(>|t|)
## (Intercept)                               26.266 2.62e-15 ***
## gapno-gap                                  6.679 1.72e-06 ***
## wh_numeric                                 2.565 0.011521 *
## modifiershort-mod                         19.612 5.72e-14 ***
## modifiermed-mod                           25.869 5.44e-16 ***
## modifierlong-mod                          15.403 3.41e-11 ***
## gapno-gap:wh_numeric                       2.130 0.034794 *
## gapno-gap:modifiershort-mod                1.470 0.143575
## gapno-gap:modifiermed-mod                  0.426 0.670639
## gapno-gap:modifierlong-mod                 0.180 0.857582
## wh_numeric:modifiershort-mod               3.602 0.000431 ***
## wh_numeric:modifiermed-mod                 4.112 6.48e-05 ***
## wh_numeric:modifierlong-mod                4.289 3.21e-05 ***
## gapno-gap:wh_numeric:modifiershort-mod     0.716 0.475241
## gapno-gap:wh_numeric:modifiermed-mod       0.536 0.592903
## gapno-gap:wh_numeric:modifierlong-mod      0.817 0.415027
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Correlation matrix not shown by default, as p = 16 > 12.
## Use print(x, correlation=TRUE)  or
##   vcov(x)     if you need it
```

```
## convergence code: 1
```

In both cases, we see the interaction we expect between wh-words and gaps, where there is significantly higher surprisal when there is no gap and a wh-word. However, (for the google model) we also see a significant interaction between the gap, wh-word and modifier. When no gap and a wh-word is present the modifiers significantly increase surprisal.

To Do – distance as a continuous variable.