

AgreementSubset

Load in the data and adjust the coding of conditions.

```
library(tidyr)
library(dplyr)

##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##   filter, lag
##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

rt.rdata <- read.csv("data/AgreementSet.csv", header=TRUE)
rt.data <- rt.rdata %>% filter(RT<=7000) %>%
  filter(ROI%in%c(-3,-2,-1,0,1,2)) %>%
  rename(participant = MD5)

rt.data$Type <- as.character(rt.data$Type) # Just in case it's automatically read as a factor

# Maybe this should all just be a recode call?
rt.data$Type[rt.data$Type == "AGREE"] <- "AGREE_G"

rt.data <- rt.data %>% separate(Type, c("Type", "pGram"), sep="_")

rt.data$pGram[rt.data$pGram == "UAMB"] <- "G"
rt.data$pGram[rt.data$pGram == "AMB"] <- "U"
rt.data$pGram[rt.data$pGram == "UNG"] <- "U"

rt.data$pGram <- as.factor(rt.data$pGram)
rt.data$Type <- as.factor(rt.data$Type)
```

Prep data for hypothesis testing.

```
# Sum coding for the position factor
rt.ht_data <- rt.data %>% subset(rt.data$ROI >= 0) # critical + spillover data only

rt.ht_data$position <- droplevels(as.factor(rt.ht_data$ROI))

contrasts(rt.ht_data$position) <- contr.sum(3)/2
contrasts(rt.ht_data$position)

##      [,1] [,2]
## 0    0.5  0.0
## 1    0.0  0.5
## 2   -0.5 -0.5
```

```
rt.ht_data$pGram.coded <- recode(rt.ht_data$pGram, "U" = 1, "G" = 0)
rt.ht_data$Type.coded <- recode(rt.ht_data$Type, "AGREE" = 0, "NPZ" = 1)
rt.ht_data$position.coded.1 <- recode(rt.ht_data$position, "0"=0.5, "1"=0, "2"=-0.5)
rt.ht_data$position.coded.2 <- recode(rt.ht_data$position, "0"=0, "1"=0.5, "2"=-0.5)
```

Correct for trial number by fitting model on fillers

```
library(lme4)
rt.fillers <- read.csv("./data/Fillers.csv") %>% filter(RT <= 7000) %>% rename(participant = MD5)

model.trialnumcorr <- lmer(RT ~ scale(trialnumber) + (1 + scale(trialnumber) | participant),
                           data=rt.fillers,
                           control=lmerControl(optimizer="bobyqa",optCtrl=list(maxfun=2e5)))

rt.ht_data$trialnum_pred <- predict(model.trialnumcorr, newdata=rt.ht_data)
rt.ht_data$RT_corrected <- rt.ht_data$RT - rt.ht_data$trialnum_pred

saveRDS(rt.ht_data, "datasets/agreement_data.rds")

rt.ht_data <- readRDS("datasets/agreement_data.rds")
summary(rt.ht_data)
```

```
##           Time                participant      Type
## Min.      :1.627e+09  0004aaa67ada7be01b5686748f11685e:  48  AGREE:44797
## 1st Qu.:1.635e+09  001a8eae4e288b7c03b45ae90394c121:  48  NPZ   :47747
## Median :1.636e+09  006bae166adc198ebfda8a6994fa19a9:  48
## Mean    :1.636e+09  007e3f4acb6480caf593aa497fccb561:  48
## 3rd Qu.:1.639e+09  00a7cad919e08678d03be9dcf98928e0:  48
## Max.    :1.640e+09  00b9b975f607a40288f26f967a4de5ba:  48
##                (Other)                :92256
## pGram      WordPosition      EachWord      EventTime
## G:46306    Min.      : 7.000  working      : 3109  Min.      :1.630e+12
## U:46238    1st Qu.: 8.000  more        : 2148  1st Qu.:1.635e+12
##           Median : 8.000  attention   : 2039  Median :1.639e+12
##           Mean   : 8.369  remains     : 1867  Mean   :1.637e+12
##           3rd Qu.: 9.000  from        : 1621  3rd Qu.:1.640e+12
##           Max.   :10.000  mysteriously: 1620  Max.   :1.640e+12
##                (Other)      :80140
##
```

```
## Whenever the nurse calls%2C the doctors stops working immediately to check on the patient.
## After the worst team loses%2C the tournament remains essentially the same for the rest of the year.
## When the lecturer stops%2C her audiences needs several minutes to reflect on the content.
## Unless the dog attacks%2C the cats remains relatively tranquil throughout the day.
## After the diplomat signs%2C the agreement creates another border conflict as a side effect.
## (Other)
## NA's
##
##           Question      Answer
## Does the nurse call the patient? : 2847  Min.      :0.0000
## Did the woman move the mail?      : 2672  1st Qu.:0.0000
## Was the position hard?             : 2651  Median :0.0000
## Did the chef join the restaurant recently? : 2646  Mean    :0.4674
## Did the guitarist complete the sound check?: 2646  3rd Qu.:1.0000
## (Other)                           :79058  Max.    :1.0000
## NA's                             :    24
```

```
##      List      item      RT_Answering      CriticalPosition consec
## j      : 5325      Min.      : 1.00      Min.      : 95      Min.      :7.000      no:92544
## f      : 5222      1st Qu.: 7.00      1st Qu.: 1722      1st Qu.:7.000
## e      : 5197      Median :12.00      Median : 2277      Median :7.000
## i      : 5187      Mean   :12.79      Mean   : 2792      Mean   :7.369
## n      : 5186      3rd Qu.:19.00      3rd Qu.: 3205      3rd Qu.:8.000
## q      : 5179      Max.    :24.00      Max.    :109053      Max.    :8.000
## (Other):61248      NA's      :24
##      CONSTRUCTION      correct      ROI      RT
## NP/Z Agreement:44797      Min.      :0.0000      Min.      :0      Min.      : 17.0
## NPZ      :47747      1st Qu.:1.0000      1st Qu.:0      1st Qu.: 279.0
##      Median :1.0000      Median :1      Median : 368.0
##      Mean   :0.8432      Mean   :1      Mean   : 451.4
##      3rd Qu.:1.0000      3rd Qu.:2      3rd Qu.: 500.0
##      Max.    :1.0000      Max.    :2      Max.    :6861.0
##
##      AMBIG      AMBUAMB      RTacross3words      trialnumber      position
## Amb :46238      Min.      :0.0000      Min.      : 30.67      Min.      : 1.00      0:30848
## Unamb:46306      1st Qu.:0.0000      1st Qu.: 302.00      1st Qu.:36.00      1:30843
##      Median :0.0000      Median : 393.33      Median :72.00      2:30853
##      Mean   :0.4996      Mean   : 459.16      Mean   :59.34
##      3rd Qu.:1.0000      3rd Qu.: 524.67      3rd Qu.:83.00
##      Max.    :1.0000      Max.    :180556.67      Max.    :92.00
##
##      pGram.coded      Type.coded      position.coded.1      position.coded.2
## Min.      :0.0000      Min.      :0.0000      Min.      :-5.0e-01      Min.      :-5.0e-01
## 1st Qu.:0.0000      1st Qu.:0.0000      1st Qu.: -5.0e-01      1st Qu.: -5.0e-01
## Median :0.0000      Median :1.0000      Median : 0.0e+00      Median : 0.0e+00
## Mean   :0.4996      Mean   :0.5159      Mean   :-2.7e-05      Mean   :-5.4e-05
## 3rd Qu.:1.0000      3rd Qu.:1.0000      3rd Qu.: 5.0e-01      3rd Qu.: 5.0e-01
## Max.    :1.0000      Max.    :1.0000      Max.    : 5.0e-01      Max.    : 5.0e-01
##
##      trialnum_pred      RT_corrected
## Min.      : 53.44      Min.      : -794.357
## 1st Qu.: 302.80      1st Qu.: -49.024
## Median : 362.33      Median : -0.521
## Mean   : 377.02      Mean   : 74.427
## 3rd Qu.: 433.71      3rd Qu.: 84.950
## Max.    :1248.02      Max.    :6445.326
##
```

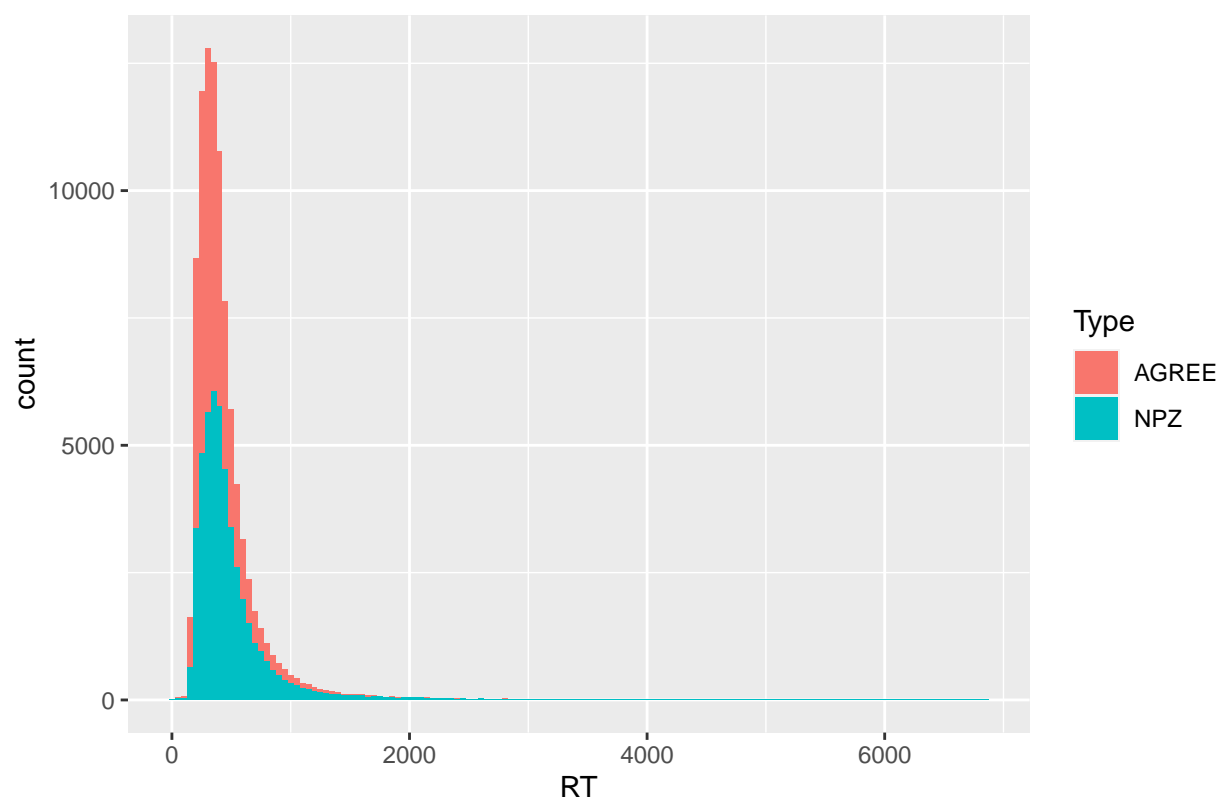
```
head(subset(rt.ht_data, is.na(rt.ht_data$pGram)))
```

```
## [1] Time      participant      Type      pGram
## [5] WordPosition      EachWord      EventTime      Sentence
## [9] Question      Answer      List      item
## [13] RT_Answering      CriticalPosition      consec      CONSTRUCTION
## [17] correct      ROI      RT      AMBIG
## [21] AMBUAMB      RTacross3words      trialnumber      position
## [25] pGram.coded      Type.coded      position.coded.1      position.coded.2
## [29] trialnum_pred      RT_corrected
## <0 rows> (or 0-length row.names)
```

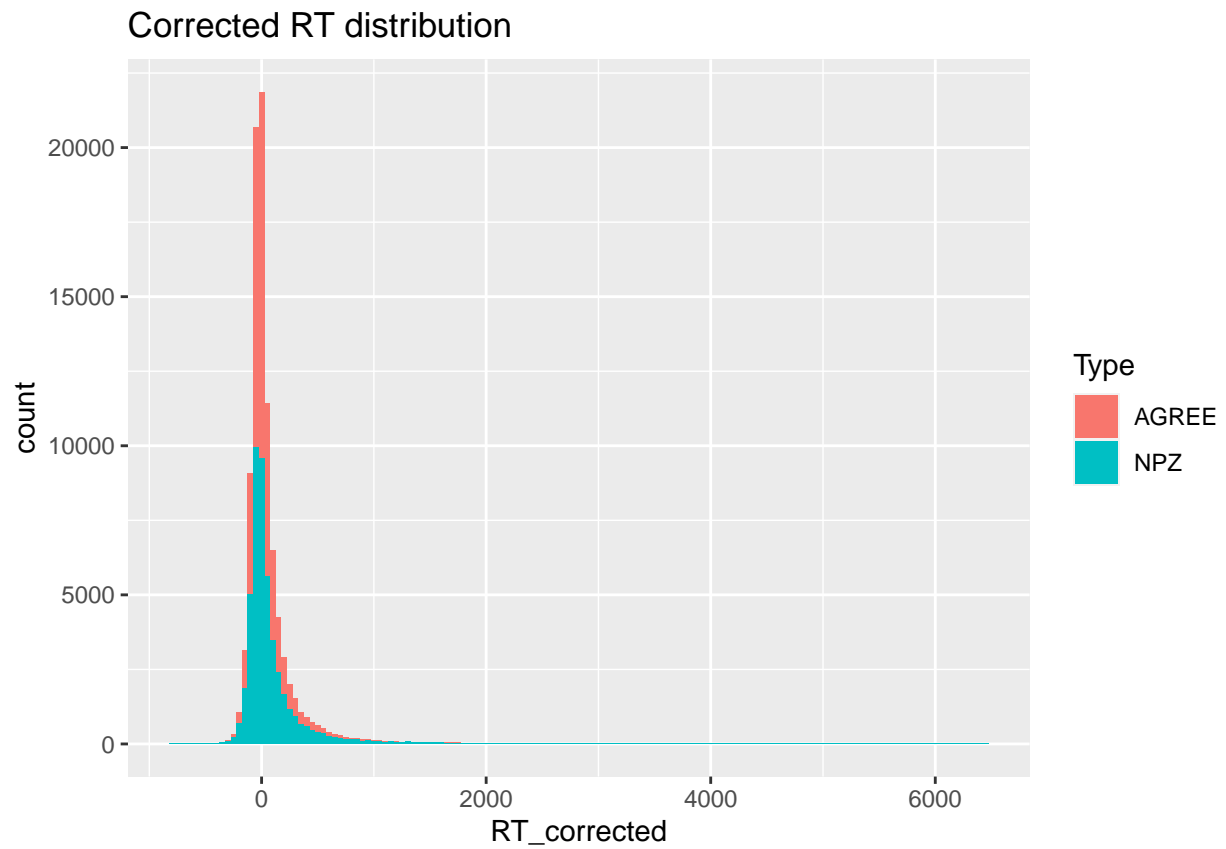
```
library(ggplot2)
```

```
ggplot(data=rt.ht_data, aes(x=RT, fill=Type)) + geom_histogram(binwidth=50) + labs(title="Raw RT distribut
```

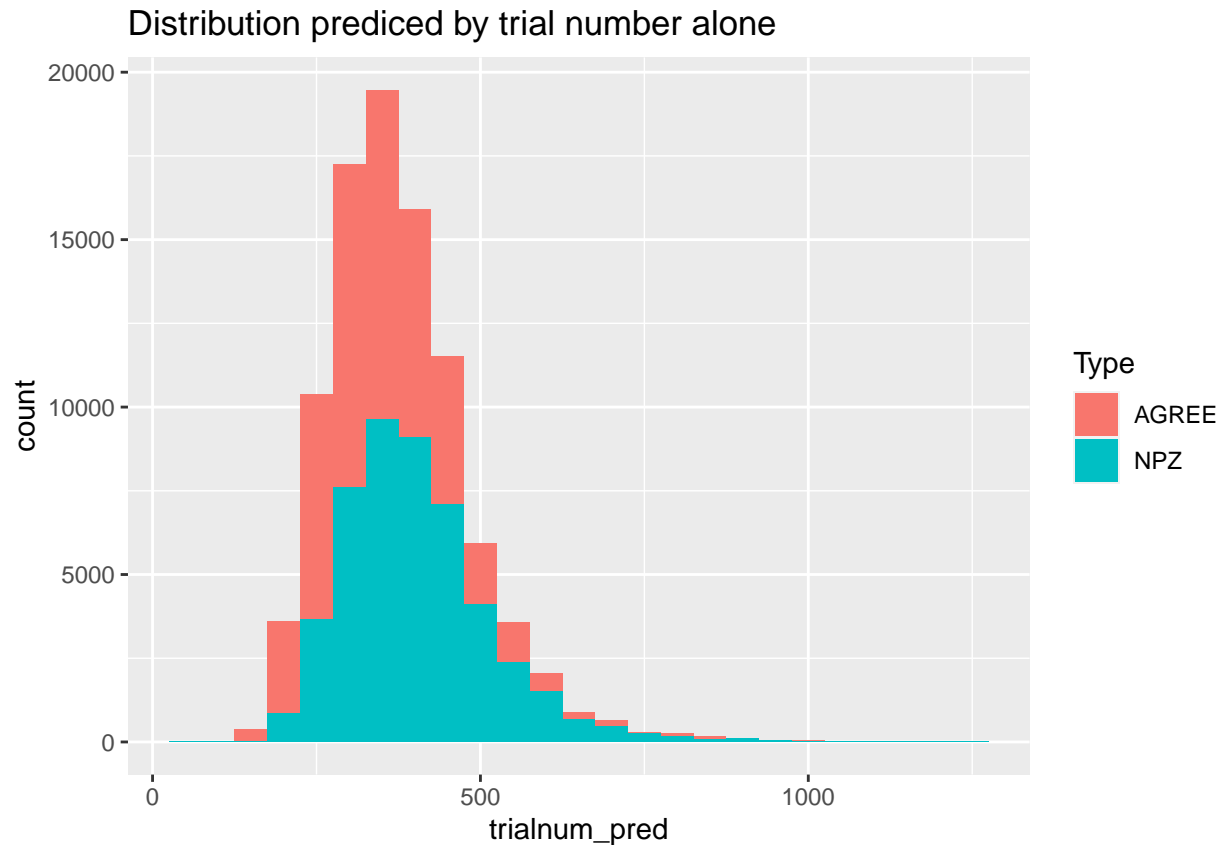
Raw RT distribution



```
ggplot(data=rt.ht_data, aes(x=RT_corrected, fill=Type)) + geom_histogram(binwidth=50) + labs(title="Corr
```



```
ggplot(data=rt.ht_data, aes(x=trialnum_pred, fill=Type)) + geom_histogram(binwidth=50) + labs(title="Dis
```



```
library(lme4)
library(lmerTest)

model.freq.prereg <- lmer(RT ~ pGram.coded * Type.coded * (position.coded.1 + position.coded.2) + (1 + pGram * Type * position | subject),
  data = rt.ht_data,
  REML = FALSE,
  control=lmerControl(optimizer="bobyqa",
    optCtrl=list(maxfun=200000)))

saveRDS(model.freq.prereg, "models/agreement_fmodel_prereg.rds")

model.freq.trialnum <- lmer(RT ~ pGram * Type * position + trialnumber + (1 + pGram * Type * position | subject),
  data = rt.ht_data,
  REML = FALSE,
  control=lmerControl(optimizer="bobyqa",
    optCtrl=list(maxfun=200000)))

saveRDS(model.freq.trialnum, "models/agreement_fmodel_trialnum.rds")
```

Bayesian Model Training Code

Prereg

```
# TO BE RERUN with explicit coding and more RES
rt.ht_data <- readRDS("datasets/agreement_data.rds")
```

```
prior1 <- c(prior("normal(300,1000)", class = "Intercept"),
  prior("normal(0,150)", class = "b"),
  prior("normal(0,200)", class = "sd"),
  prior("normal(0,500)", class = "sigma"))

rt.bmodel <- brm(RT | trunc(lb=0) ~ Type * pGram * position + (1 + Type * pGram + position || item) + (
  data = rt.ht_data,
  prior = prior1,
  iter = 4000,
  cores = 4,
  seed = 117,
)

saveRDS(rt.bmodel, "models/agreement_bmodel_prior1.rds")
```

Corrected by predictor

```
rt.ht_data <- readRDS("datasets/agreement_data.rds")

prior1 <- c(prior("normal(300,1000)", class = "Intercept"),
  prior("normal(0,150)", class = "b"),
  prior("normal(0,200)", class = "sd"),
  prior("normal(0,500)", class = "sigma"))

rt.bmodel <- brm(RT | trunc(lb=0) ~ Type.coded * pGram.coded * (position.coded.1 + position.coded.2) + (
  data = rt.ht_data,
  prior = prior1,
  iter = 6000,
  cores = 4,
  seed = 117,
)

saveRDS(rt.bmodel, "models/agreement_bmodel_prior1_trialnum.max.rds")
```

Corrected by residualizing

```
rt.ht_data <- readRDS("datasets/agreement_data.rds")

prior1 <- c(prior("normal(300,1000)", class = "Intercept"),
  prior("normal(0,150)", class = "b"),
  prior("normal(0,200)", class = "sd"),
  prior("normal(0,500)", class = "sigma"))

rt.bmodel <- brm(RT_corrected ~ Type.coded * pGram.coded * (position.coded.1 + position.coded.2) + (1 +
  data = rt.ht_data,
  prior = prior1,
  iter = 6000,
  cores = 4,
  seed = 117,
)
```

```
saveRDS(rt.bmodel, "models/agreement_bmodel_prior1_corrected.rds")
```

Load models in

```
library(brms)
model.prior1.prereg <- readRDS("models/agreement_bmodel_prior1.rds")
summary(model.prior1.prereg)

model.prior1.trialnum <- readRDS("models/agreement_bmodel_prior1_trialnum.max.rds")
summary(model.prior1.trialnum)

model.prior1.corrected <- readRDS("models/agreement_bmodel_prior1_corrected.rds")
summary(model.prior1.trialnum)
```

Plot HDIs

Note: A little to memory intensive to run local, even one at a time.

```
library(bayestestR)
library(ggplot2)
library(dplyr)
library(stringr)

model.prior1.prereg.hdi <- hdi(model.prior1.prereg, ci=0.95, effects="all")

ggplot(subset(model.prior1.prereg.hdi, str_detect(Parameter, 'b_')) +
  geom_linerange(mapping=aes(x = Parameter,
                             ymin = CI_low,
                             ymax = CI_high)) +
  geom_hline(yintercept=0, linetype="dashed") +
  theme_classic() + theme(axis.text.x=element_text(angle=45, hjust=1)))

model.prior1.trialnum.hdi <- hdi(model.prior1.trialnum, ci=0.95, effects="all")

ggplot(subset(model.prior1.trialnum.hdi, str_detect(Parameter, 'b_')) +
  geom_linerange(mapping=aes(x = Parameter,
                             ymin = CI_low,
                             ymax = CI_high)) +
  geom_hline(yintercept=0, linetype="dashed") +
  theme_classic() + theme(axis.text.x=element_text(angle=45, hjust=1)))

model.prior1.corrected.hdi <- hdi(model.prior1.corrected, ci=0.95, effects="all")

ggplot(subset(model.prior1.corrected.hdi, str_detect(Parameter, 'b_')) +
  geom_linerange(mapping=aes(x = Parameter,
                             ymin = CI_low,
                             ymax = CI_high)) +
  geom_hline(yintercept=0, linetype="dashed") +
  theme_classic() + theme(axis.text.x=element_text(angle=45, hjust=1)))
```


Get posterior samples

```
posterior.prereg <- posterior_samples(model.prior1.prereg)
posterior.prereg <- subset(posterior.prereg, select=!str_detect(colnames(posterior.prereg), "r_particip

posterior.trialnum <- posterior_samples(model.prior1.trialnum)
posterior.trialnum <- subset(posterior.trialnum, select=!str_detect(colnames(posterior.trialnum), "r_pa

posterior.corrected <- posterior_samples(model.prior1.corrected)
posterior.corrected <- subset(posterior.corrected, select=!str_detect(colnames(posterior.corrected), "r_
```

Plots by position in critical region

```
# Effect size by position

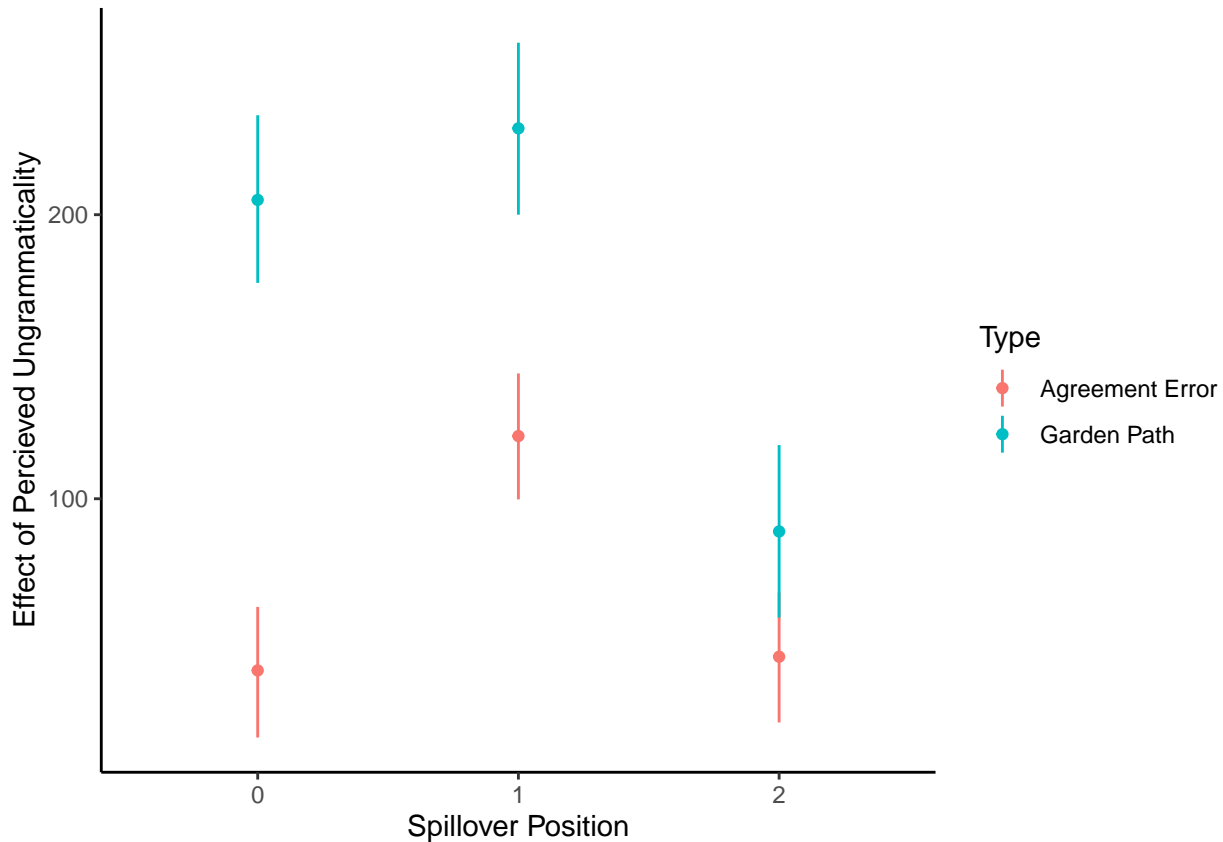
bypos_analysis <- function(posterior) {
  fixed_effs <- posterior %>%
    mutate(GPE_0 = b_pGramU + `b_TypeNPZ:pGramU` + 0.5 * `b_pGramU:position1` + 0.5 * `b_Type
           GPE_1 = b_pGramU + `b_TypeNPZ:pGramU` + 0.5 * `b_pGramU:position2` + 0.5 * `b_Type
           GPE_2 = b_pGramU + `b_TypeNPZ:pGramU` - 0.5 * `b_pGramU:position1` - 0.5 * `b_Type
           Agr_0 = b_pGramU + 0.5 * `b_pGramU:position1`,
           Agr_1 = b_pGramU + 0.5 * `b_pGramU:position2`,
           Agr_2 = b_pGramU - 0.5 * `b_pGramU:position1` - 0.5 * `b_pGramU:position2`) %>%
    gather("effect", "size", c("GPE_0", "GPE_1", "GPE_2", "Agr_0", "Agr_1", "Agr_2")) %>%
    group_by(effect) %>%
    summarize(mean=mean(size),
              lower = quantile(size, 0.025)[[1]],
              upper = quantile(size, 0.975)[[1]]) %>%
    separate("effect", c("type", "position"), sep = "_")
  return(fixed_effs)
}

saveRDS(bypos_analysis(posterior.prereg), "datasets/agreement.prereg.by_pos.rds")

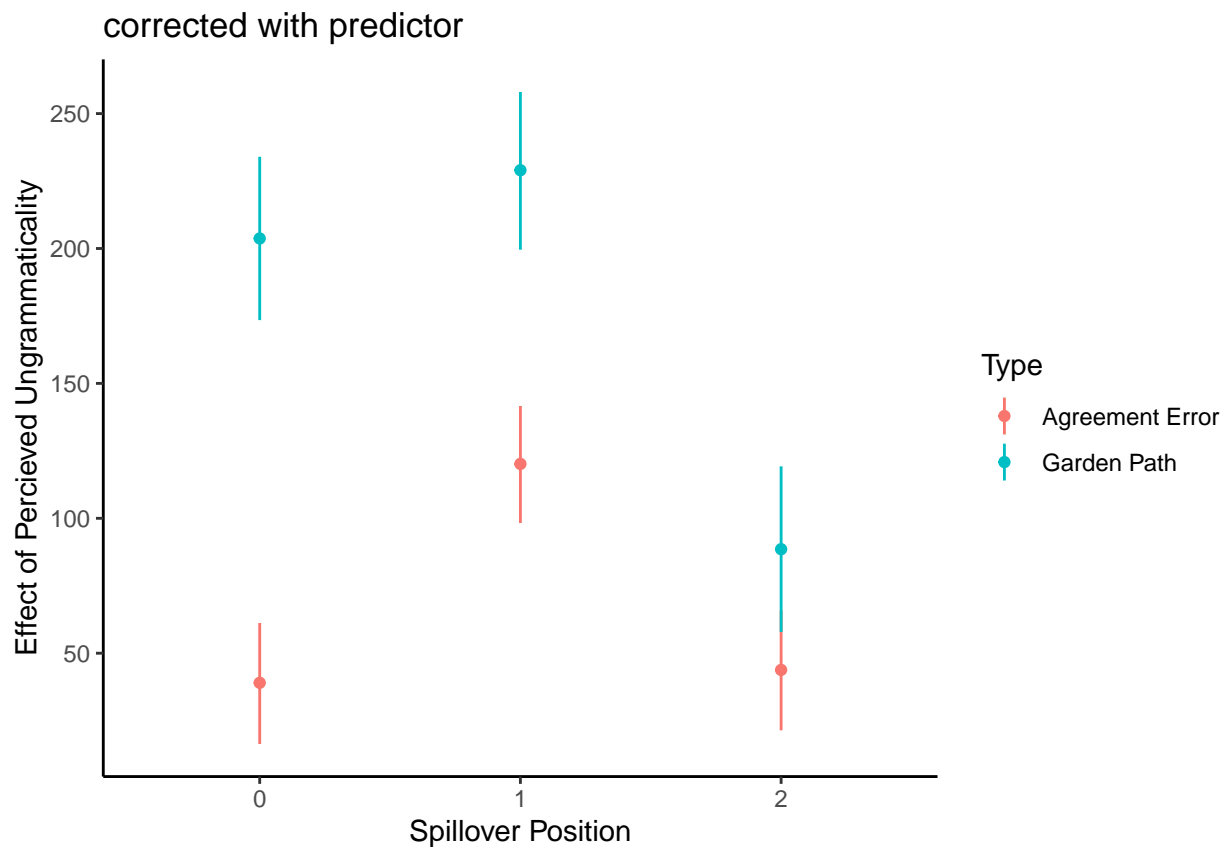
bypos_analysis.corr <- function(posterior) {
  fixed_effs <- posterior %>%
    mutate(GPE_0 = b_pGram.coded + `b_Type.coded:pGram.coded` + 0.5 * `b_pGram.coded:position
           GPE_1 = b_pGram.coded + `b_Type.coded:pGram.coded` + 0.5 * `b_pGram.coded:position
           GPE_2 = b_pGram.coded + `b_Type.coded:pGram.coded` - 0.5 * `b_pGram.coded:position
           Agr_0 = b_pGram.coded + 0.5 * `b_pGram.coded:position.coded.1`,
           Agr_1 = b_pGram.coded + 0.5 * `b_pGram.coded:position.coded.2`,
           Agr_2 = b_pGram.coded - 0.5 * `b_pGram.coded:position.coded.1` - 0.5 * `b_pGram.co
    gather("effect", "size", c("GPE_0", "GPE_1", "GPE_2", "Agr_0", "Agr_1", "Agr_2")) %>%
    group_by(effect) %>%
    summarize(mean=mean(size),
              lower = quantile(size, 0.025)[[1]],
              upper = quantile(size, 0.975)[[1]]) %>%
    separate("effect", c("type", "position"), sep = "_")
  return(fixed_effs)
}

#saveRDS(bypos_analysis.corr(posterior.trialnum), "datasets/agreement.trialnum.by_pos.rds")
#saveRDS(bypos_analysis.corr(posterior.corrected), "datasets/agreement.corrected.by_pos.rds")
```

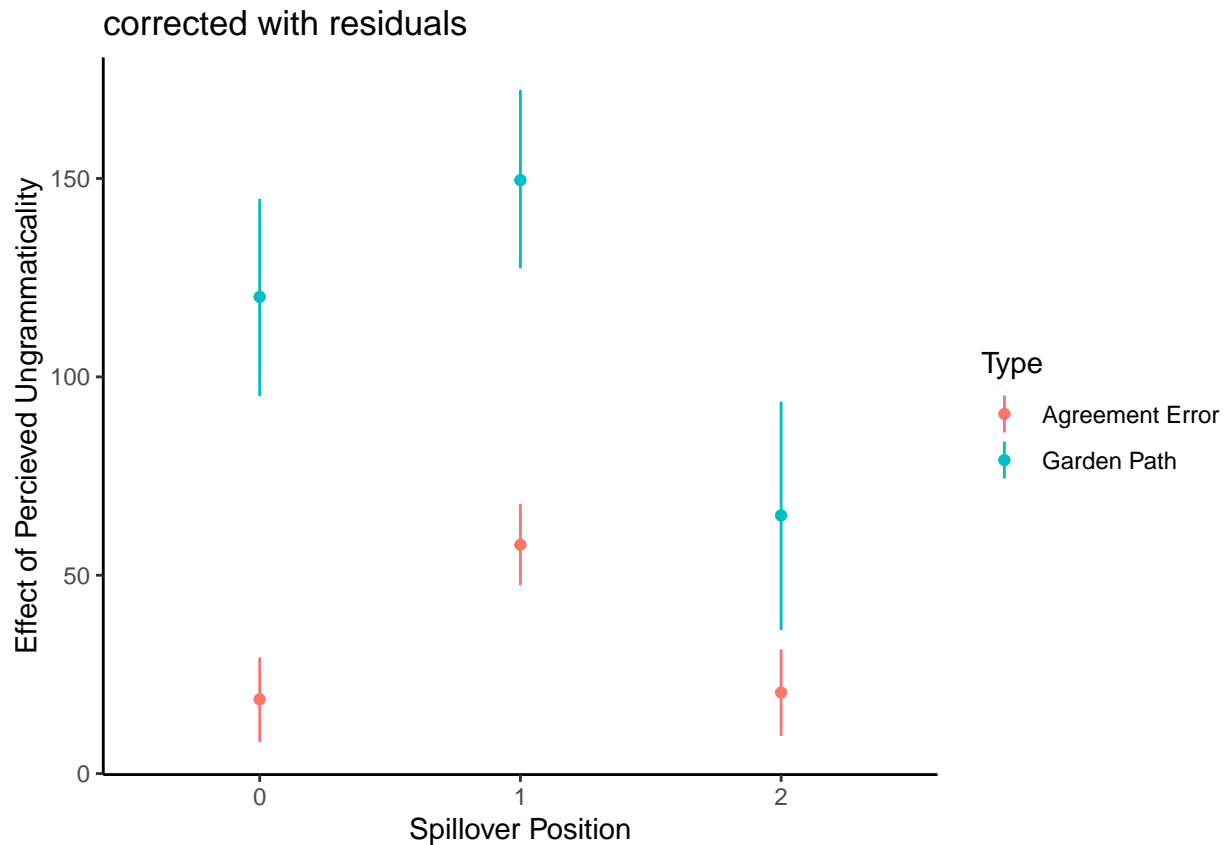
```
fixed_effs.prereg <- readRDS("datasets/agreement.prereg.by_pos.rds")
# Plotting
ggplot(fixed_effs.prereg, aes(x=position, color=type, y=mean)) +
  geom_linerange(mapping=aes(ymin=lower,
                             ymax=upper)) +
  geom_point() +
  scale_color_discrete(labels=c("Agr"="Agreement Error", "GPE"="Garden Path")) +
  theme_classic() + labs(x="Spillover Position", y="Effect of Percieved Ungrammaticality", color="Type")
```



```
fixed_effs.trialnum <- readRDS("datasets/agreement.trialnum.by_pos.rds")
# Plotting
ggplot(fixed_effs.trialnum, aes(x=position, color=type, y=mean)) +
  geom_linerange(mapping=aes(ymin=lower,
                             ymax=upper)) +
  geom_point() +
  scale_color_discrete(labels=c("Agr"="Agreement Error", "GPE"="Garden Path")) +
  theme_classic() + labs(x="Spillover Position", y="Effect of Percieved Ungrammaticality", color="Type")
```



```
fixed_effs.corrected <- readRDS("datasets/agreement.corrected.by_pos.rds")
# Plotting
ggplot(fixed_effs.corrected, aes(x=position, color=type, y=mean)) +
  geom_linerange(mapping=aes(ymin=lower,
                             ymax=upper)) +
  geom_point() +
  scale_color_discrete(labels=c("Agr"="Agreement Error", "GPE"="Garden Path")) +
  theme_classic() + labs(x="Spillover Position", y="Effect of Percieved Ungrammaticality", color="Type")
```



Plots by item

Note: effects averaged over the full critical region

```
library(reshape2)

# By Item
byitem_analysis <- function(posterior) {
  print(colnames(posterior))
  cns = c()
  ran_eff <- posterior
  for(i in 1:24) {
    gram_name <- paste0("r_item[", toString(i), ",pGramU]")
    inter_name <- paste0("r_item[", i, ",TypeNPZ:pGramU]")
    ran_eff <- ran_eff %>% mutate("GPE_{i}" := b_pGramU + `b_TypeNPZ:pGramU` + !!as.name(gram_name) + !
                                "Agr_{i}" := b_pGramU + !!as.name(gram_name))
    cns <- append(cns, c(paste0("GPE_", toString(i)), paste0("Agr_", toString(i))))
  }

  ran_eff <- ran_eff %>% gather("item", "effect", all_of(cns)) %>%
    group_by(item) %>%
    summarize(mean=mean(effect),
              lower = quantile(effect, 0.025)[[1]],
              upper = quantile(effect, 0.975)[[1]]) %>%
    separate("item", c("type", "item_num"), sep = "_")
  return(ran_eff)
}
```

```

saveRDS(byitem_analysis(posterior.trialnum), "datasets/agreement.trialnum.by_item.rds")

# By Item
byitem_analysis.corr <- function(posterior) {
  print(colnames(posterior))
  cns = c()
  ran_eff <- posterior
  for(i in 1:24) {
    gram_name <- paste0("r_item[", toString(i), "pGram.coded]")
    inter_name <- paste0("r_item[", i, ",Type.coded:pGram.coded]")
    ran_eff <- ran_eff %>% mutate("GPE_{i}" := `b_pGram.coded` + `b_Type.coded:pGram.coded` + !!as.name(
      "Agr_{i}" := `b_pGram.coded` + !!as.name(gram_name))
    cns <- append(cns, c(paste0("GPE_", toString(i)), paste0("Agr_", toString(i))))
  }

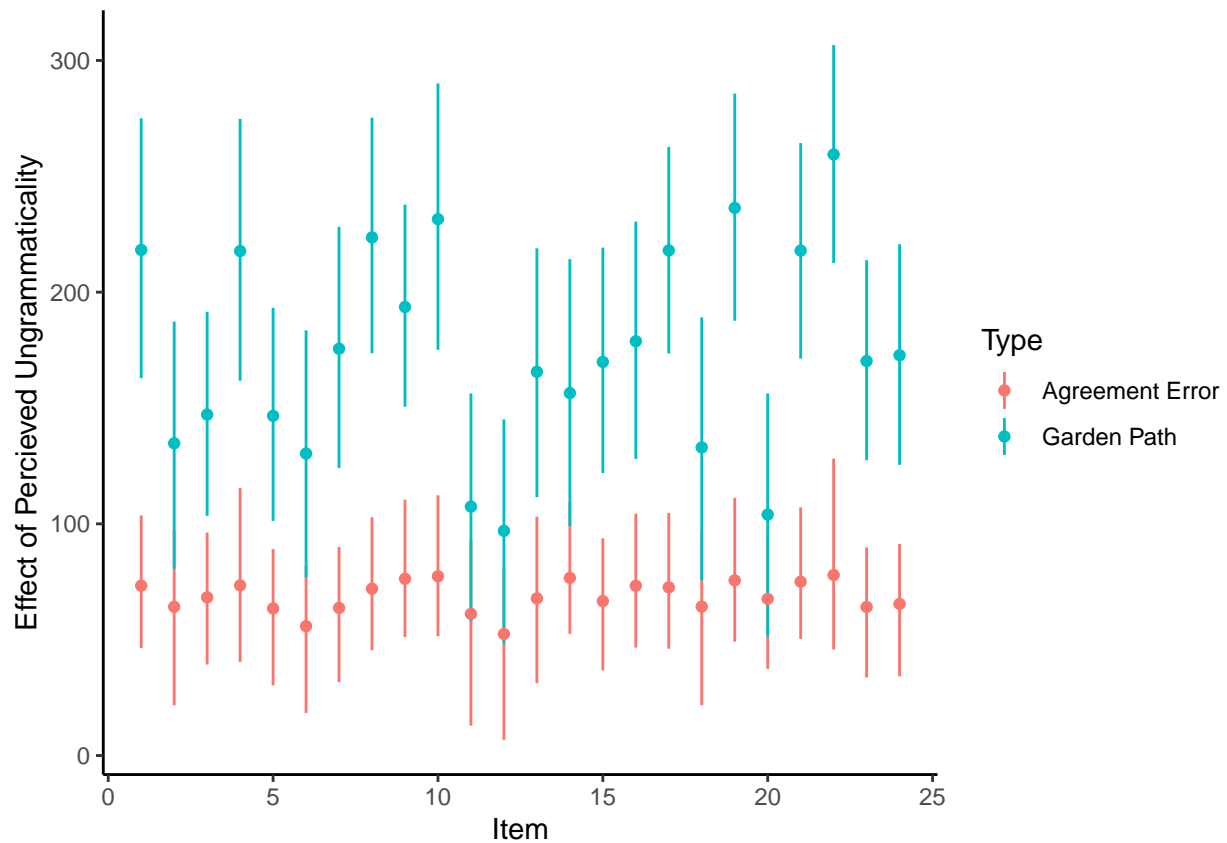
  ran_eff <- ran_eff %>% gather("item", "effect", all_of(cns)) %>%
    group_by(item) %>%
    summarize(mean=mean(effect),
              lower = quantile(effect, 0.025)[[1]],
              upper = quantile(effect, 0.975)[[1]]) %>%
    separate("item", c("type", "item_num"), sep = "_")
  return(ran_eff)
}

saveRDS(byitem_analysis.corr(posterior.prereg), "datasets/agreement.prereg.by_item.rds")
saveRDS(byitem_analysis.corr(posterior.corrected), "datasets/agreement.corrected.by_item.rds")

ran_effs <- readRDS("datasets/agreement.prereg.by_item.rds")

ggplot(ran_effs, aes(x=as.numeric(item_num), color=type, y=mean)) +
  geom_linerange(mapping=aes(ymin=lower,
                             ymax=upper)) +
  geom_point() +
  scale_color_discrete(labels=c("Agr"="Agreement Error", "GPE"="Garden Path")) +
  theme_classic() + labs(x="Item", y="Effect of Percieved Ungrammaticality", color="Type")

```

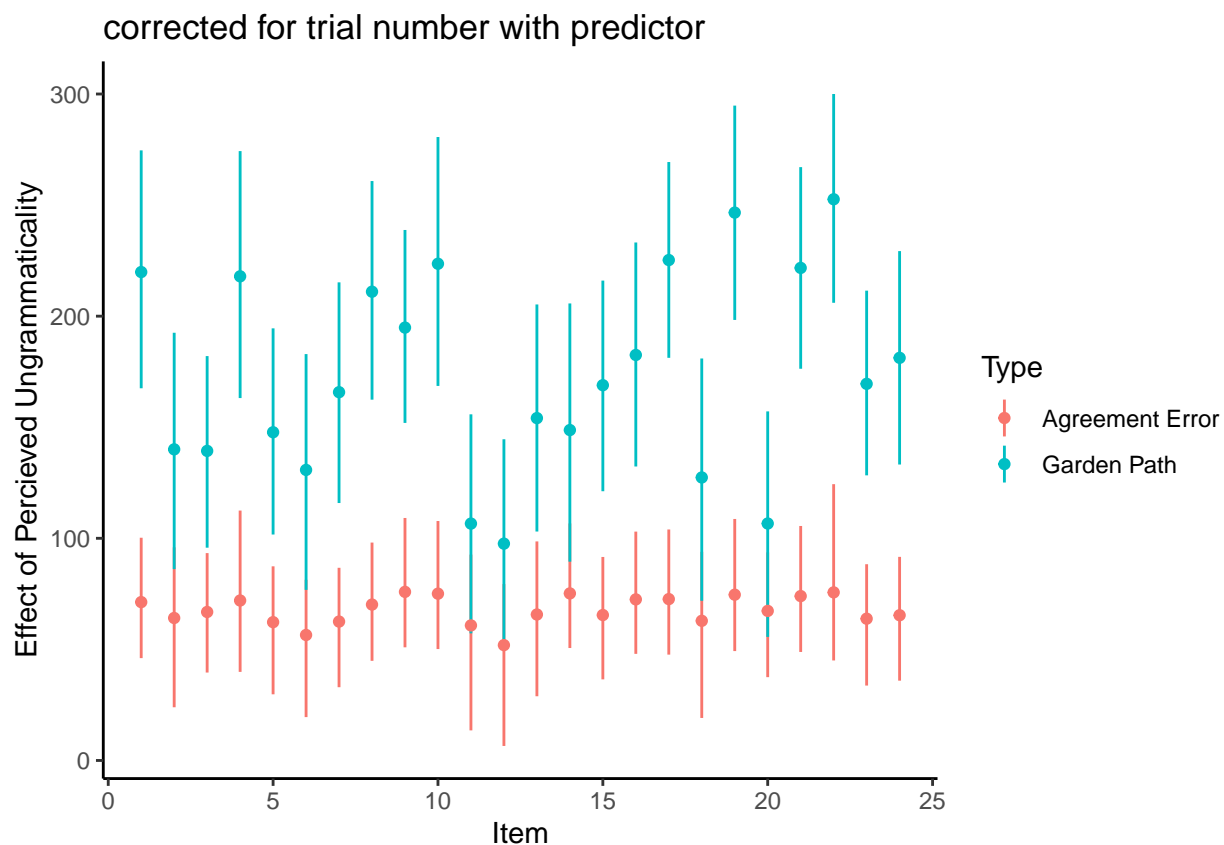


```

ran_effs <- readRDS("datasets/agreement.trialnum.by_item.rds")

ggplot(ran_effs, aes(x=as.numeric(item_num), color=type, y=mean)) +
  geom_linerange(mapping=aes(ymin=lower,
                             ymax=upper)) +
  geom_point() +
  scale_color_discrete(labels=c("Agr"="Agreement Error", "GPE"="Garden Path")) +
  theme_classic() + labs(x="Item", y="Effect of Percieved Ungrammaticality", color="Type", title="corre

```

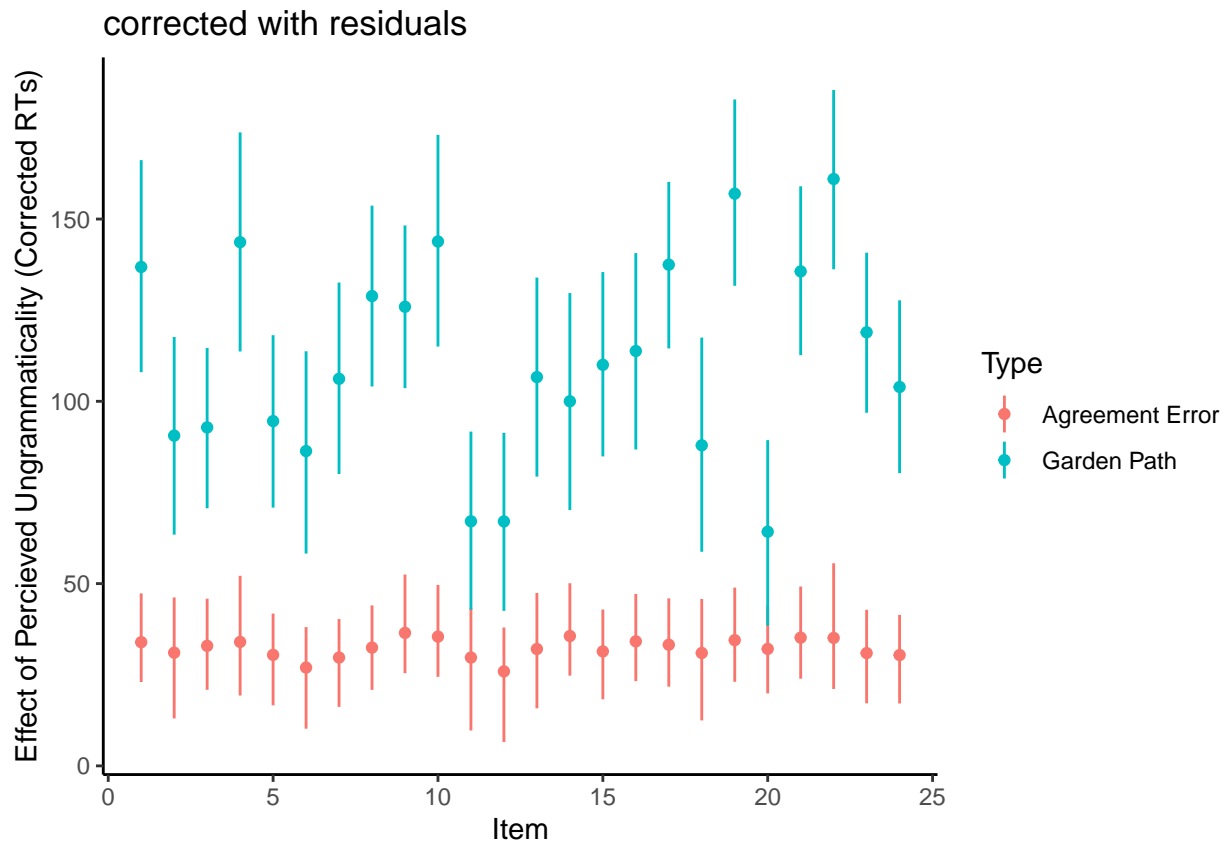


```

ran_effs <- readRDS("datasets/agreement.corrected.by_item.rds")

ggplot(ran_effs, aes(x=as.numeric(item_num), color=type, y=mean)) +
  geom_linerange(mapping=aes(ymin=lower,
                             ymax=upper)) +
  geom_point() +
  scale_color_discrete(labels=c("Agr"="Agreement Error", "GPE"="Garden Path")) +
  theme_classic() + labs(x="Item", y="Effect of Percieved Ungrammaticality (Corrected RTs)", color="Type")

```



Reformat data so it's averaged by group and by position

Note: testing this right now to see if I can compute this and work with this within memory constraints off of MARCC. Useful for surprisal correlations

```
# By Item and position
byboth_analysis <- function(posterior) {
  print(colnames(posterior))
  cns = c()
  ran_eff <- posterior
  for(i in 1:24) {
    gram_name <- paste0("r_item[", toString(i), ",pGramU]")
    inter_name <- paste0("r_item[", toString(i), ",TypeNPZ:pGramU]")
    ran_eff <- ran_eff %>% mutate("GPE_{i}" := b_pGramU + `b_TypeNPZ:pGramU` + !!as.name(gram_name) + !
                                "Agr_{i}" := b_pGramU + !!as.name(gram_name))

    gpe_avg <- paste0("GPE_", toString(i))
    agr_avg <- paste0("Agr_", toString(i))

    # For RE structure (1 + pGram * Type + position // item) - update as needed if using a more rich RE
    ran_eff <- ran_eff %>% mutate("GPE_{i}_0" := !!as.name(gpe_avg) + 0.5 * `b_pGramU:position1` + 0.5 *
                                "GPE_{i}_1" := !!as.name(gpe_avg) + 0.5 * `b_pGramU:position2` + 0.5 *
                                "GPE_{i}_2" := !!as.name(gpe_avg) - 0.5 * `b_pGramU:position1` - 0.5 *
                                "Agr_{i}_0" := !!as.name(agr_avg) + 0.5 * `b_pGramU:position1`,
                                "Agr_{i}_1" := !!as.name(agr_avg) + 0.5 * `b_pGramU:position2`,
                                "Agr_{i}_2" := !!as.name(agr_avg) - 0.5 * `b_pGramU:position2` - 0.5 *
                                )
  }
}
```



```

    for(i in 0:2) {
      cns <- append(cns, c(paste0("GPE_", toString(i), "_", toString(j)), paste0("Agr_", toString(i), "_", toString(j))))
    }
  }

  ran_eff <- ran_eff %>% gather("item", "effect", all_of(cns)) %>%
    group_by(item) %>%
    summarize(mean=mean(effect),
              lower = quantile(effect, 0.025)[[1]],
              upper = quantile(effect, 0.975)[[1]]) %>%
    separate("item", c("type", "item_num", "position"), sep = "_")
  return(ran_eff)
}

saveRDS(byboth_analysis(posterior.prereg), "datasets/agreement.prereg.by_both.rds")

# By Item and position
byboth_analysis.cor <- function(posterior) {
  print(colnames(posterior))
  cns = c()
  ran_eff <- posterior
  for(i in 1:24) {
    gram_name <- paste0("r_item[", toString(i), "pGram.coded]")
    inter_name <- paste0("r_item[", toString(i), "TypeNPZ:pGram.coded]")
    ran_eff <- ran_eff %>% mutate("GPE_{i}" := b_pGram.coded + `b_Type.coded:pGram.coded` + !!as.name(gram_name)
                                "Agr_{i}" := b_pGram.coded + !!as.name(gram_name))

    gpe_avg <- paste0("GPE_", toString(i))
    agr_avg <- paste0("Agr_", toString(i))

    # For RE structure (1 + pGram * Type * position // item) - update as needed if using a more rich RE
    pos1_name <- paste0("r_item[", toString(i), "pGram.coded:position.coded.1]")
    pos2_name <- paste0("r_item[", toString(i), "pGram.coded:position.coded.2]")
    pos1_inter_name <- paste0("r_item[", toString(i), "Type.coded:pGram.coded:position.coded.1]")
    pos2_inter_name <- paste0("r_item[", toString(i), "Type.coded:pGram.coded:position.coded.2]")

    ran_eff <- ran_eff %>% mutate("GPE_{i}_0" := !!as.name(gpe_avg) + 0.5 * `b_pGram.coded:position.coded.1`
                                "GPE_{i}_1" := !!as.name(gpe_avg) + 0.5 * `b_pGram.coded:position.coded.2`
                                "GPE_{i}_2" := !!as.name(gpe_avg) - 0.5 * `b_pGram.coded:position.coded.1`
                                "Agr_{i}_0" := !!as.name(agr_avg) + 0.5 * `b_pGram.coded:position.coded.1`
                                "Agr_{i}_1" := !!as.name(agr_avg) + 0.5 * `b_pGram.coded:position.coded.2`
                                "Agr_{i}_2" := !!as.name(agr_avg) - 0.5 * `b_pGram.coded:position.coded.1`
                                )

    for(i in 0:2) {
      cns <- append(cns, c(paste0("GPE_", toString(i), "_", toString(j)), paste0("Agr_", toString(i), "_", toString(j))))
    }
  }

  ran_eff <- ran_eff %>% gather("item", "effect", all_of(cns)) %>%
    group_by(item) %>%
    summarize(mean=mean(effect),
              lower = quantile(effect, 0.025)[[1]],
              upper = quantile(effect, 0.975)[[1]]) %>%
    separate("item", c("type", "item_num", "position"), sep = "_")
  return(ran_eff)
}

```

```
}

saveRDS(byboth_analysis.corr(posterior.trialnum), "datasets/agreement.trialnum.by_both.rds")
saveRDS(byboth_analysis.corr(posterior.corrected), "datasets/agreement.corrected.by_both.rds")
```

Correlations with Surprisal

```
# Surprisal Correlations
library(ggplot2)
surps.lstm <- readRDS("datasets/agreement_data.lstm.rds")

surps.lstm.corrplot <- surps.lstm %>% gather("surp_p", "surprisal", c("surprisal_lstm", "surprisal_lstm",
  group_by(item, Type, pGram, surp_p) %>%
  summarize(surprisal=mean(surprisal)) %>%
  spread(pGram, surprisal) %>%
  mutate(ugram_eff = U-G)

## `summarise()` has grouped output by 'item', 'Type', 'pGram'. You can override
## using the `.groups` argument.

ran_effs <- readRDS("datasets/agreement.trialnum.by_item.rds")
ran_effs

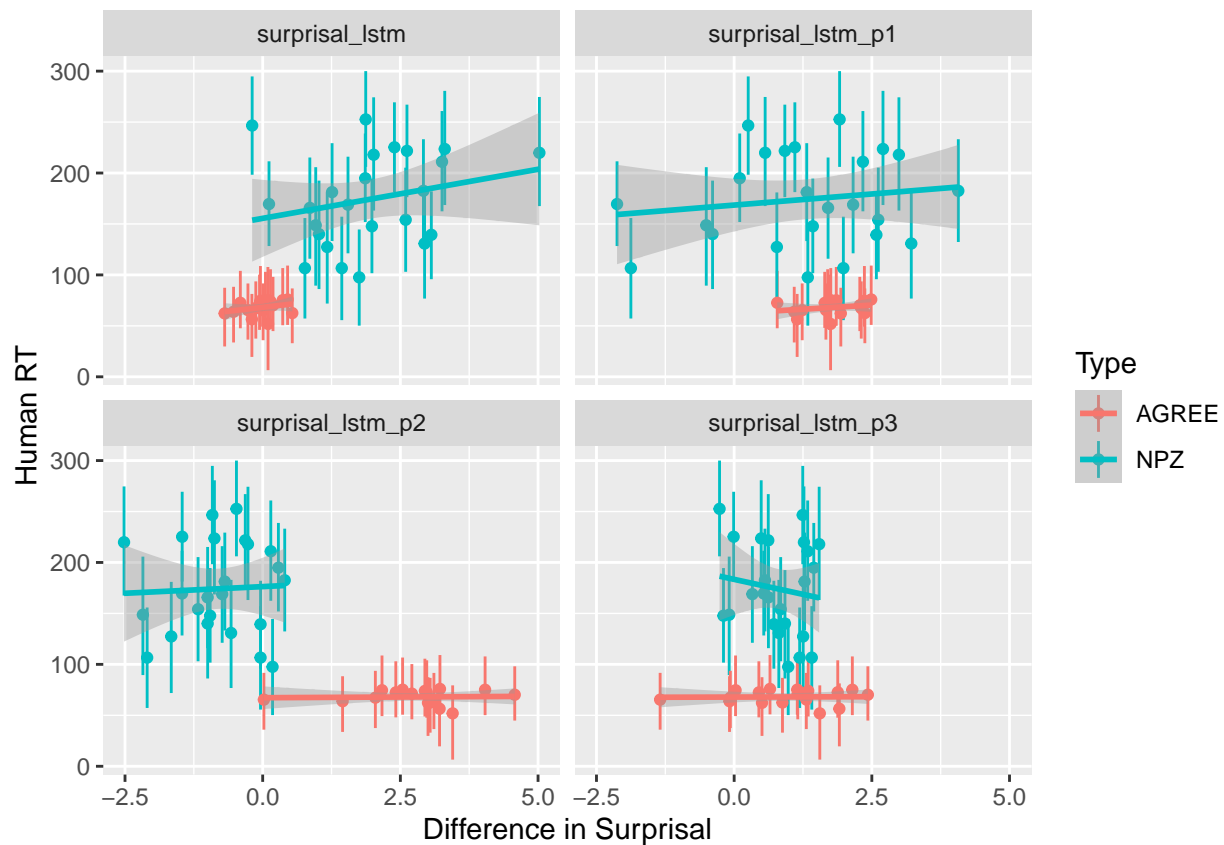
## # A tibble: 48 x 5
##   type item_num mean lower upper
##   <chr> <chr>   <dbl> <dbl> <dbl>
## 1 Agr    1      71.3  46.1  100.
## 2 Agr   10      75.1  50.2  108.
## 3 Agr   11      60.8  13.5   92.7
## 4 Agr   12      51.9   6.53  79.4
## 5 Agr   13      65.7  28.9   98.6
## 6 Agr   14      75.2  50.6  107.
## 7 Agr   15      65.5  36.5   91.6
## 8 Agr   16      72.5  48.0  103.
## 9 Agr   17      72.6  47.6  104.
## 10 Agr  18      62.9  19.1   93.9
## # ... with 38 more rows

ran_effs$type <- recode(ran_effs$type, "Agr" = "AGREE", "GPE" = "NPZ")

surps.lstm.corrplot <- merge(x=surps.lstm.corrplot, y=ran_effs, by.x=c("item", "Type"), by.y=c("item_num", "Type"))

ggplot(data=surps.lstm.corrplot, aes(x=ugram_eff, y=mean, color=Type)) + geom_linerange(aes(ymin=lower, ymax=upper))

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



```
surps.gpt <- readRDS("datasets/agreement_data.gpt2.rds")

surps.gpt.corrplot <- surps.gpt %>% gather("surp_p", "surprisal", c("surprisal_gpt2", "surprisal_gpt2_p"))
  group_by(item, Type, pGram, surp_p) %>%
  summarize(surprisal=mean(surprisal)) %>%
  spread(pGram, surprisal) %>%
  mutate(ugram_eff = U-G)
```

```
## `summarise()` has grouped output by 'item', 'Type', 'pGram'. You can override
## using the `.groups` argument.
```

```
ran_effs <- readRDS("datasets/agreement.trialnum.by_item.rds")
ran_effs
```

```
## # A tibble: 48 x 5
##   type item_num mean lower upper
##   <chr> <chr>   <dbl> <dbl> <dbl>
## 1 Agr    1      71.3  46.1  100.
## 2 Agr   10      75.1  50.2  108.
## 3 Agr   11      60.8  13.5   92.7
## 4 Agr   12      51.9   6.53  79.4
## 5 Agr   13      65.7  28.9  98.6
## 6 Agr   14      75.2  50.6  107.
## 7 Agr   15      65.5  36.5  91.6
## 8 Agr   16      72.5  48.0  103.
## 9 Agr   17      72.6  47.6  104.
## 10 Agr  18      62.9  19.1  93.9
## # ... with 38 more rows
```

```

ran_effs$type <- recode(ran_effs$type, "Agr" = "AGREE", "GPE" = "NPZ")

surps.gpt.corrplot <- merge(x=surps.gpt.corrplot, y=ran_effs, by.x=c("item", "Type"), by.y=c("item_num", "Type"))

ggplot(data=surps.gpt.corrplot, aes(x=ugram_eff, y=mean, color=Type)) + geom_linerange(aes(ymin=lower, ymax=upper))

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

```

