Mandarin

* This has a set of analyses on Mandarin RCs for this paper. Yang, W., Chan, A., Chang, F., & Kidd, E. Four-year-old Mandarin-speaking children’s online comprehension of relative clauses.
* First we load in the looking data and plot looks to target in proportions for the whole test period.

## permutation script 2019  
require(ggplot2)

## Loading required package: ggplot2

require(lme4)

## Loading required package: lme4

## Loading required package: Matrix

require(stringr)

## Loading required package: stringr

require(nlme) ## for lme()

## Loading required package: nlme

##   
## Attaching package: 'nlme'

## The following object is masked from 'package:lme4':  
##   
## lmList

require(emmeans) ## for lsmeans()

## Loading required package: emmeans

# may need to run this command if the following doesn work install.packages("multcomp")  
require(multcomp) ## for multiple comparison stuff

## Loading required package: multcomp

## Loading required package: mvtnorm

## Loading required package: survival

## Loading required package: TH.data

## Loading required package: MASS

##   
## Attaching package: 'TH.data'

## The following object is masked from 'package:MASS':  
##   
## geyser

##   
## Attaching package: 'multcomp'

## The following object is masked from 'package:emmeans':  
##   
## cld

require(remef)

## Loading required package: remef

#Set your working directory  
# Go to Session in menu, go to Set Working Directory,   
# click on To Source File Location  
print(sessionInfo())

## R version 3.5.2 (2018-12-20)  
## Platform: x86\_64-apple-darwin15.6.0 (64-bit)  
## Running under: macOS Mojave 10.14.6  
##   
## Matrix products: default  
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib  
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib  
##   
## locale:  
## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] remef\_1.0.6.9000 multcomp\_1.4-8 TH.data\_1.0-9 MASS\_7.3-51.1   
## [5] survival\_2.43-3 mvtnorm\_1.0-8 emmeans\_1.3.1 nlme\_3.1-137   
## [9] stringr\_1.3.1 lme4\_1.1-21 Matrix\_1.2-15 ggplot2\_3.1.0   
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_1.0.0 nloptr\_1.2.1 pillar\_1.3.1 compiler\_3.5.2   
## [5] plyr\_1.8.4 bindr\_0.1.1 tools\_3.5.2 boot\_1.3-20   
## [9] digest\_0.6.18 evaluate\_0.12 tibble\_2.0.1 gtable\_0.2.0   
## [13] lattice\_0.20-38 pkgconfig\_2.0.2 rlang\_0.3.1 yaml\_2.2.0   
## [17] xfun\_0.4 bindrcpp\_0.2.2 coda\_0.19-2 withr\_2.1.2   
## [21] dplyr\_0.7.8 knitr\_1.21 grid\_3.5.2 tidyselect\_0.2.5  
## [25] glue\_1.3.0 R6\_2.3.0 rmarkdown\_1.11 minqa\_1.2.4   
## [29] purrr\_0.2.5 magrittr\_1.5 codetools\_0.2-16 scales\_1.0.0   
## [33] htmltools\_0.3.6 splines\_3.5.2 assertthat\_0.2.0 xtable\_1.8-3   
## [37] colorspace\_1.3-2 sandwich\_2.5-0 estimability\_1.3 stringi\_1.2.4   
## [41] lazyeval\_0.2.1 munsell\_0.5.0 crayon\_1.3.4 zoo\_1.8-4

apaformat <- function(p){  
 p = p + theme\_bw() # make the theme black-and-white rather than grey (do this before font changes, or it overrides them)  
 p = p + theme(  
# plot.title = theme\_text(face="bold", size=14), # use theme\_get() to see available options  
# axis.title.x = theme\_text(face="bold", size=12),  
# axis.title.y = theme\_text(face="bold", size=12, angle=90),  
 panel.grid.major = element\_blank(), # switch off major gridlines  
 panel.grid.minor = element\_blank(), # switch off minor gridlines  
# legend.position = c(0.2,0.8), # manually position the legend (numbers being from 0,0 at bottom left of whole plot to 1,1 at top right)  
# legend.title = theme\_blank(), # switch off the legend title  
# legend.text = theme\_text(size=12),  
# legend.key.size = unit(1.5, "lines"),  
 legend.key = element\_blank() # switch off the rectangle around symbols in the legend  
 )  
 return(p)  
}  
  
  
modelComparisonPrint<-function(mlist){  
 for (i in 2:length(mlist)){  
 am1=anova(mlist[[i-1]],mlist[[i]])  
 terms = attr(terms(mlist[[i-1]]),"term.labels")  
 print(am1)  
 sig = ifelse(am1$`Pr(>Chisq)`[2]<0.05," \*\*\* ","")  
 print(paste("########## Above comparison for ",terms[length(terms)],sig))  
 }  
}  
  
modelComparison <- function(model,modellist=list(),verbose=0){  
 # print(model)  
 terms = attr(terms(model),"term.labels")  
 modellist = append(modellist,model)  
 # print(terms)  
 if (length(terms) > 0){  
 newformula = paste(". ~ . - ",terms[length(terms)],"")  
 print(paste("remove",newformula))  
 model2 = update(model, as.formula(newformula))  
 if (verbose > 0){  
 print(summary(model2))  
 }  
 am1=anova(model2, model)  
 print(am1)  
 sig = ifelse(am1$`Pr(>Chisq)`[2]<0.05," \*\*\* ","")  
 print(paste("########## Above comparison for ",terms[length(terms)],sig))  
   
 terms = attr(terms(model),"term.labels")  
 modellist = modelComparison(model2,modellist)  
 }  
   
 return(modellist)  
}  
  
  
getModelNumHighestTerm <- function(varname,modellist) {  
 for (i in 1:length(modellist)) {  
 terms = attr(terms(modellist[[i]]), "term.labels")  
   
 if (length(terms) > 0 & terms[length(terms)] == varname) {  
 return(i)  
 }  
 }  
 return(-1)  
}  
  
#cat(paste(printMixedModelResults(modellist),collapse="\n"))  
  
roundNonZero <- function(val){  
 tval = val  
 pow = 1;  
 repeat {  
 if (abs(tval) > 1){  
 return(round(val,pow))  
 }  
 tval = tval \* 10  
 if (pow == 5){  
 return(round(val,pow))  
 }  
# print(tval)  
# print(pow)  
 pow = pow + 1  
 }  
 return(val)  
}  
  
report <- function(varname, mlist){  
 omnimodel = mlist[[1]]  
 model.sum = summary(omnimodel)  
 model.coef = coefficients(model.sum)  
 modelnames= rev(rownames(model.coef))  
 modelnum = which(modelnames==varname)  
 m.anova = anova(mlist[[modelnum]],mlist[[modelnum+1]])  
 chisq = round(m.anova$Chisq[2],2)  
# pval = roundNonZero(m.anova$`Pr(>Chisq)`[2])  
 pval = format.pval(pv = m.anova$`Pr(>Chisq)`[2], digits = 2,eps=0.001,nsmall = 3)  
   
 oneline = model.coef[varname,]  
 beta = roundNonZero(oneline[1])  
 se = round(oneline[2],2)  
   
 outstr = paste("$\\beta$=",beta,", SE=",se,", $\\chi^2$(1)=",chisq,", p=",pval,sep="")  
 outstr = gsub("=<","<",outstr)  
 return(outstr)  
}  
  
logit2prob <- function(logit){  
 odds <- exp(logit)  
 prob <- odds / (1 + odds)  
 return(prob)  
}  
  
printPostHoc<-function(post){  
 beta = roundNonZero(post$test$coefficients)  
 z= roundNonZero(post$test$tstat)  
 pval = post$test$pvalues  
 pval = format.pval(pv = post$test$pvalues, digits = 2,eps=0.001,nsmall = 3)  
  
 outstr = paste("$\\beta$=",beta,", z=",z,", p=",pval,sep="")  
 outstr = gsub("=<","<",outstr)  
 return(outstr)  
}  
  
printPostHocEM<-function(posthocs,dim,val){  
 r = which(posthocs[[dim]]==val)  
 z= roundNonZero(posthocs$z.ratio[r])  
 pval = posthocs$p.value[r]  
 pval = format.pval(pv = pval, digits = 2,eps=0.001,nsmall = 3)  
  
 outstr = paste("z=",z,", p=",pval,sep="")  
 outstr = gsub("=<","<",outstr)  
 return(outstr)  
}  
  
# data frame with onset for each region  
onsetStimCL = data.frame(SentenceType=c(rep("Subject",4),rep("Object",5)),cat = c("VN","DE","DCL","Head","N","V","DE","DCL","Head"), onset=c(0,0.808,1.02,1.49,0,0.431,0.767,1.027,1.427))  
onsetStimCL$HeadNoun = "DCL"  
onsetStimDE = data.frame(SentenceType=c(rep("Subject",3),rep("Object",4)),cat = c("VN","DE","Head","N","V","DE","Head"), onset=c(0,0.797,0.972,0,0.493,0.821,0.963))  
onsetStimDE$HeadNoun = "DE"  
onsetStim = rbind(onsetStimCL,onsetStimDE)  
onsetStim$time = onsetStim$onset \* 1000  
#onsetStim$time2=c(onsetStim$time[2:length(onsetStim$time)],0)  
#onsetStim$time2[onsetStim$cat=="Head"] = 1000\*c(2.001,1.876,1.479,1.501)  
onsetStim$time2 = c(805,1017,1487,2001, #subject DCL offsets  
 428,764,1024,1424,1876, # object DCL  
 794,969,1479, # subject DE  
 490, 818,960,1501) # object DE  
onsetStim$ypos = 0.95  
onsetStim$ypos[onsetStim$SentenceType == "Object"] = 0.75  
onsetStim$target = 1  
onsetStim$restarg = 0  
onsetStim[onsetStim$cat=="Head",]

## SentenceType cat onset HeadNoun time time2 ypos target restarg  
## 4 Subject Head 1.490 DCL 1490 2001 0.95 1 0  
## 9 Object Head 1.427 DCL 1427 1876 0.75 1 0  
## 12 Subject Head 0.972 DE 972 1479 0.95 1 0  
## 16 Object Head 0.963 DE 963 1501 0.75 1 0

aggregate(time2 ~ HeadNoun + cat, onsetStim, mean)

## HeadNoun cat time2  
## 1 DCL DCL 1455.5  
## 2 DCL DE 1020.5  
## 3 DE DE 964.5  
## 4 DCL Head 1938.5  
## 5 DE Head 1490.0  
## 6 DCL N 428.0  
## 7 DE N 490.0  
## 8 DCL V 764.0  
## 9 DE V 818.0  
## 10 DCL VN 805.0  
## 11 DE VN 794.0

bothdataAll=read.csv("ManRC eyetracking data-4yrs-final.csv")  
bothdata=bothdataAll  
xtabs(~ Participant,bothdata)

## Participant  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18   
## 732 671 610 488 671 793 732 671 671 305 793 549 549 549 732 732 488 610   
## 19 20 21 22 23   
## 793 366 549 793 671

bothdata$HeadNoun <- factor(bothdata$HeadNoun,labels=c("DCL","DE"))  
bothdata$SentenceType=factor(bothdata$SentenceType,labels=c("Subject","Object"))  
  
bothdata$Item = str\_replace(bothdata$Item,"(S|O)","I")  
bothdata$time = as.numeric(as.character(bothdata$time))  
bothdata$time = bothdata$time \* 1000  
bothdata$target = bothdata$T  
#/(both$T+both$D)  
xtabs(~ Participant + HeadNoun , bothdata)

## HeadNoun  
## Participant DCL DE  
## 1 366 366  
## 2 366 305  
## 3 305 305  
## 4 244 244  
## 5 366 305  
## 6 427 366  
## 7 305 427  
## 8 244 427  
## 9 366 305  
## 10 0 305  
## 11 366 427  
## 12 305 244  
## 13 244 305  
## 14 305 244  
## 15 305 427  
## 16 427 305  
## 17 244 244  
## 18 305 305  
## 19 427 366  
## 20 366 0  
## 21 305 244  
## 22 488 305  
## 23 305 366

xtabs(~ Participant + SentenceType , bothdata)

## SentenceType  
## Participant Subject Object  
## 1 427 305  
## 2 427 244  
## 3 183 427  
## 4 305 183  
## 5 244 427  
## 6 305 488  
## 7 305 427  
## 8 366 305  
## 9 366 305  
## 10 183 122  
## 11 427 366  
## 12 488 61  
## 13 366 183  
## 14 305 244  
## 15 488 244  
## 16 427 305  
## 17 427 61  
## 18 183 427  
## 19 427 366  
## 20 183 183  
## 21 305 244  
## 22 427 366  
## 23 305 366

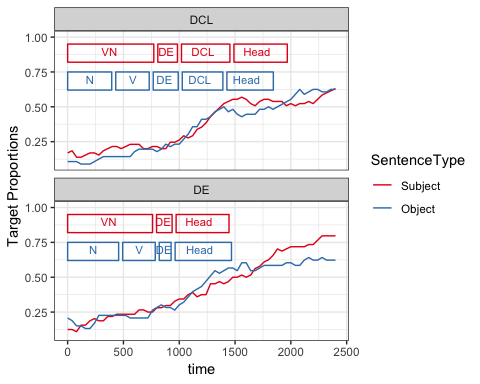
xtabs(~ Item + HeadNoun , bothdata)

## HeadNoun  
## Item DCL DE  
## I1 1464 0  
## I2 2074 0  
## I3 1891 0  
## I4 1952 0  
## I5 0 1647  
## I6 0 1952  
## I7 0 1647  
## I8 0 1891

xtabs(~ Item + SentenceType, bothdata)

## SentenceType  
## Item Subject Object  
## I1 732 732  
## I2 915 1159  
## I3 1159 732  
## I4 1159 793  
## I5 854 793  
## I6 1037 915  
## I7 915 732  
## I8 1098 793

means.df = aggregate(cbind(target) ~ time + HeadNoun + SentenceType, bothdata, mean)  
p = ggplot(means.df , aes( x = time, y = target, colour=SentenceType))  
p = p + geom\_line()  
p = p + scale\_colour\_brewer(palette="Set1")  
p = p + theme\_bw() # change background to white  
p = p + ylab("Target Proportions")  
#p = p + ylim(0,1)  
p = p + facet\_wrap(~ HeadNoun, ncol=1)  
p = p + geom\_rect(data=onsetStim,aes(xmin=time,xmax = time2-33, ymin=ypos-0.13, ymax=ypos,colour=SentenceType),fill=NA, show.legend=FALSE)  
p = p + geom\_text(data=onsetStim,aes(x=(time+time2)/2, y=ypos-0.05, label=cat),hjust=0.7,size=3,show.legend=FALSE)  
p



Above is the raw data, but next we will do a mixed model analysis using the empirical logit. We zero the empirical logit at time 0 to remove any earlier preferences. Since children could be doing prediction or have biases for particular structures before the sentence disambiguate, we do an analysis on all of the data.

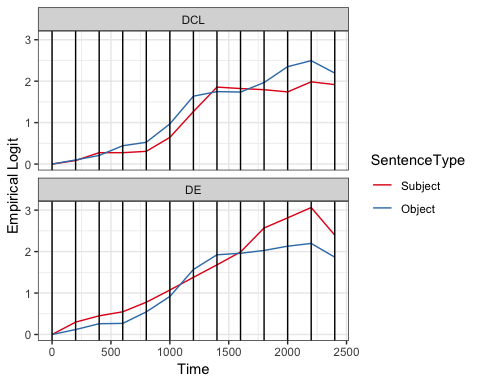
winsize = 200  
  
bothdata$win = floor(bothdata$time/winsize) # like floor  
xtabs(~ win, bothdata)

## win  
## 0 1 2 3 4 5 6 7 8 9 10 11 12   
## 1190 1190 1190 1190 1190 1190 1190 1190 1190 1190 1190 1190 238

#xtabs(~ Item + SentenceType+ win, both)  
#xtabs(~ Participant + SentenceType, both)  
bothdata$frames = 1  
  
bothdata$frames = 1  
subjsum.df = aggregate(cbind(target,frames) ~ win + HeadNoun + SentenceType + Participant + Item, bothdata, sum)  
subjsum.df$elog <- log( (subjsum.df$target + .5) / (subjsum.df$frames - subjsum.df$target + .5) )  
subjsum.df$time=subjsum.df$win\*winsize  
xtabs(~ time,subjsum.df)

## time  
## 0 200 400 600 800 1000 1200 1400 1600 1800 2000 2200 2400   
## 238 238 238 238 238 238 238 238 238 238 238 238 238

# remove preference at time 0  
meansubset = subset(subjsum.df,time == 0)  
meanpartitemdf = aggregate(elog ~ Participant + Item + HeadNoun + SentenceType, meansubset, mean)  
both3 = merge(subjsum.df,meanpartitemdf, all.x=TRUE,by=c("Participant","Item","HeadNoun","SentenceType"),sort=F)  
both3$elog = both3$elog.x-both3$elog.y  
  
# plot figure for dataset  
means.df = aggregate(cbind(elog) ~ time + HeadNoun + SentenceType, both3, mean)  
p = ggplot(means.df , aes( x = time, y = elog, colour=SentenceType))  
p = p + geom\_line()  
p = p + scale\_colour\_brewer(palette="Set1")  
p = p + theme\_bw() # change background to white  
p = p + ylab("Empirical Logit")  
p = p + xlab("Time")  
#p = p + ylim(0,1)  
p = p + facet\_wrap(~ HeadNoun, ncol=1)  
p+geom\_vline(xintercept = 200\*0:12)



ggsave("elogitall.png")

## Saving 5 x 4 in image

Here is a mixed model analysis using 200 ms windows

subjmeans.df = both3  
subjmeans.df$cwin = subjmeans.df$win - mean(subjmeans.df$win)  
subjmeans.df$cCL = ifelse(subjmeans.df$HeadNoun == "DCL",0.5,-0.5)  
subjmeans.df$cobject = ifelse(subjmeans.df$SentenceType == "Object",0.5,-0.5)  
subjmeans.df$Participant=factor(subjmeans.df$Participant)  
subjmeans.df$Item=factor(subjmeans.df$Item)  
  
mixmodel = lmer(elog ~ cwin\*cCL\*cobject + (1 | Participant) + (1 | Item), subjmeans.df)  
summixed=summary(mixmodel)  
print(summixed)

## Linear mixed model fit by REML ['lmerMod']  
## Formula: elog ~ cwin \* cCL \* cobject + (1 | Participant) + (1 | Item)  
## Data: subjmeans.df  
##   
## REML criterion at convergence: 13609.9  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.4991 -0.5887 -0.1405 0.7565 2.5025   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## Participant (Intercept) 0.18753 0.433   
## Item (Intercept) 0.02789 0.167   
## Residual 4.64737 2.156   
## Number of obs: 3094, groups: Participant, 23; Item, 8  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 1.222930 0.115070 10.628  
## cwin 0.222674 0.010397 21.417  
## cCL -0.141131 0.142682 -0.989  
## cobject -0.109515 0.080912 -1.354  
## cwin:cCL -0.021361 0.020794 -1.027  
## cwin:cobject -0.007539 0.020794 -0.363  
## cCL:cobject 0.470619 0.158184 2.975  
## cwin:cCL:cobject 0.079111 0.041589 1.902  
##   
## Correlation of Fixed Effects:  
## (Intr) cwin cCL cobjct cwn:CL cwn:cb cCL:cb  
## cwin 0.000   
## cCL -0.002 0.000   
## cobject 0.035 0.000 -0.005   
## cwin:cCL 0.000 -0.018 0.000 0.000   
## cwin:cobjct 0.000 0.084 0.000 0.000 -0.011   
## cCL:cobject -0.004 0.000 0.049 -0.002 0.000 0.000   
## cwn:cCL:cbj 0.000 -0.011 0.000 0.000 0.084 -0.018 0.000

print(unique(subjmeans.df$win))

## [1] 0 1 2 3 4 5 6 7 8 9 10 11 12

modellist = modelComparison(mixmodel)

## [1] "remove . ~ . - cwin:cCL:cobject "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + cCL + cobject + (1 | Participant) + (1 | Item) +   
## model2: cwin:cCL + cwin:cobject + cCL:cobject  
## model: elog ~ cwin \* cCL \* cobject + (1 | Participant) + (1 | Item)  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)   
## model2 10 13600 13660 -6790.1 13580   
## model 11 13598 13665 -6788.2 13576 3.6233 1 0.05698 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## [1] "########## Above comparison for cwin:cCL:cobject "  
## [1] "remove . ~ . - cCL:cobject "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + cCL + cobject + (1 | Participant) + (1 | Item) +   
## model2: cwin:cCL + cwin:cobject  
## model: elog ~ cwin + cCL + cobject + (1 | Participant) + (1 | Item) +   
## model: cwin:cCL + cwin:cobject + cCL:cobject  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)   
## model2 9 13607 13661 -6794.5 13589   
## model 10 13600 13660 -6790.1 13580 8.8709 1 0.002897 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## [1] "########## Above comparison for cCL:cobject \*\*\* "  
## [1] "remove . ~ . - cwin:cobject "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + cCL + cobject + (1 | Participant) + (1 | Item) +   
## model2: cwin:cCL  
## model: elog ~ cwin + cCL + cobject + (1 | Participant) + (1 | Item) +   
## model: cwin:cCL + cwin:cobject  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)  
## model2 8 13605 13653 -6794.6 13589   
## model 9 13607 13661 -6794.5 13589 0.107 1 0.7436  
## [1] "########## Above comparison for cwin:cobject "  
## [1] "remove . ~ . - cwin:cCL "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + cCL + cobject + (1 | Participant) + (1 | Item)  
## model: elog ~ cwin + cCL + cobject + (1 | Participant) + (1 | Item) +   
## model: cwin:cCL  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)  
## model2 7 13604 13647 -6795.3 13590   
## model 8 13605 13653 -6794.6 13589 1.4253 1 0.2325  
## [1] "########## Above comparison for cwin:cCL "  
## [1] "remove . ~ . - cobject "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + cCL + (1 | Participant) + (1 | Item)  
## model: elog ~ cwin + cCL + cobject + (1 | Participant) + (1 | Item)  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)  
## model2 6 13604 13640 -6796.1 13592   
## model 7 13604 13647 -6795.3 13590 1.7446 1 0.1866  
## [1] "########## Above comparison for cobject "  
## [1] "remove . ~ . - cCL "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + (1 | Participant) + (1 | Item)  
## model: elog ~ cwin + cCL + (1 | Participant) + (1 | Item)  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)  
## model2 5 13604 13634 -6796.9 13594   
## model 6 13604 13640 -6796.1 13592 1.455 1 0.2277  
## [1] "########## Above comparison for cCL "  
## [1] "remove . ~ . - cwin "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ (1 | Participant) + (1 | Item)  
## model: elog ~ cwin + (1 | Participant) + (1 | Item)  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)   
## model2 4 14032 14056 -7011.9 14024   
## model 5 13604 13634 -6796.9 13594 430.15 1 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## [1] "########## Above comparison for cwin \*\*\* "

There was a main effect of window, =0.22, SE=0.01, (1)=430.15, p<0.001, and an interaction of Headnoun and sentence type =0.47, SE=0.16, (1)=8.87, p=0.0029. Since we have marginal three way interaction of HeadNoun, SentenceType, and window =0.079, SE=0.04, (1)=3.62, p=0.057, we can do posthocs. We use emmeans to do contrasts between obj and sub for each window in each HeadNoun. emmeans also adjusts p automatically for the number of comparisons.

options(contrasts = c("contr.treatment", "contr.poly"))  
# may need to run this command if the following doesn work install.packages("nlme")  
subjmeans.df$win = factor(subjmeans.df$win)  
subjmeans.df$HeadNoun = factor(subjmeans.df$HeadNoun)  
subjmeans.df$SentenceType = factor(subjmeans.df$SentenceType)  
mixmodelFactor = lmer(target ~ win\*HeadNoun\*SentenceType + (1 | Participant) + (1 | Item), subjmeans.df)  
#print(summary(mixmodelFactor))  
#save(mixmodelFactor,"mixmodelSentSlopeFactor.RData")  
model.lsmobj <- emmeans(mixmodelFactor, ~ SentenceType | HeadNoun\* win)

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, set emm\_options(pbkrtest.limit = 3094) or larger,  
## but be warned that this may result in large computation time and memory use.

## Note: D.f. calculations have been disabled because the number of observations exceeds 3000.  
## To enable adjustments, set emm\_options(lmerTest.limit = 3094) or larger,  
## but be warned that this may result in large computation time and memory use.

posthocs = summary(as.glht(pairs(model.lsmobj)))  
print(posthocs)

## $`HeadNoun = DCL, win = 0`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.3511 0.3691 0.951 0.342  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 0`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 -0.03451 0.37545 -0.092 0.927  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 1`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.3363 0.3691 0.911 0.362  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 1`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.1270 0.3754 0.338 0.735  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 2`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.4291 0.3691 1.162 0.245  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 2`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.1668 0.3754 0.444 0.657  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 3`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.1791 0.3691 0.485 0.628  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 3`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.2951 0.3754 0.786 0.432  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 4`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.1206 0.3691 0.327 0.744  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 4`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.1931 0.3754 0.514 0.607  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 5`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.01015 0.36913 0.028 0.978  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 5`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.1438 0.3754 0.383 0.702  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 6`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 -0.00688 0.36913 -0.019 0.985  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 6`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 -0.2418 0.3754 -0.644 0.52  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 7`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.4835 0.3691 1.31 0.19  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 7`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 -0.2689 0.3754 -0.716 0.474  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 8`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.4398 0.3691 1.192 0.233  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 8`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.04037 0.37545 0.108 0.914  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 9`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.2077 0.3691 0.563 0.574  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 9`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.5554 0.3754 1.479 0.139  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 10`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 -0.2824 0.3691 -0.765 0.444  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 10`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)   
## Subject - Object == 0 0.6954 0.3754 1.852 0.064 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 11`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 -0.1459 0.3691 -0.395 0.693  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 11`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)   
## Subject - Object == 0 0.8824 0.3754 2.35 0.0188 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DCL, win = 12`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.07224 0.36913 0.196 0.845  
## (Adjusted p values reported -- single-step method)  
##   
##   
## $`HeadNoun = DE, win = 12`  
##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(>|z|)  
## Subject - Object == 0 0.2980 0.3754 0.794 0.427  
## (Adjusted p values reported -- single-step method)

# only one window is significant $`HeadNoun = DE, win = 11`  
  
subjmeans.df$pred <- remef(mixmodelFactor, ran = "all")  
means2.df = aggregate(cbind(pred,target) ~ time + HeadNoun + SentenceType, subjmeans.df, mean)  
timelist = as.integer(unique(subjmeans.df$time))  
p = ggplot(means2.df , aes( x = time, y = target, colour=SentenceType))  
meanwin= data.frame(time= rep(timelist,2),HeadNoun=rep(c("DCL","DE"),each=length(timelist)))  
#meanwin$time = (meanwin$win-1)\*winsize  
meanwin$pval = as.numeric(lapply(posthocs, function(x){ return( x$test$pvalues[1]) }) )  
meanwin$target = 0  
meanwin$pred =0  
meanwin$SentenceType = "Object"  
meansigPost = subset(meanwin,pval < 0.05)   
# color them grey

The mixed model above yields one window that shows a significant difference (highlighted in red).

Now we do the same empirical logit mixed model analysis using DE offset as the start time. This analysis assumes that children have no structural biases or predictions before DE offset.

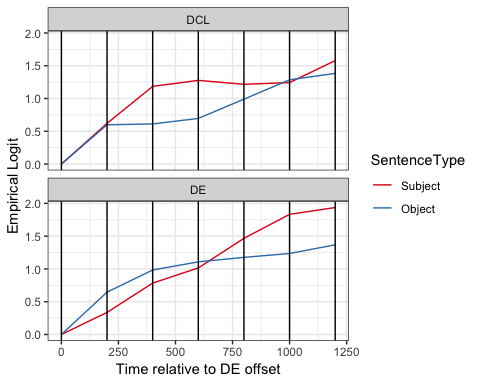
offsetDE = read.csv("DEoffset.csv")  
both2 = merge(bothdata,offsetDE, all.x=TRUE,by=c("Item","SentenceType"),sort=F)  
both2$time = both2$time - both2$Offset\*1000  
#xtabs(~ time, both2)  
  
winsize = 200  
  
both2$win = floor(both2$time/winsize) # like floor  
xtabs(~ win, both2)

## win  
## -6 -5 -4 -3 -2 -1 0 1 2 3 4 5 6 7   
## 183 1039 1190 1190 1190 1190 1190 1190 1190 1190 1190 1190 1093 303

#xtabs(~ Item + SentenceType+ win, both)  
#xtabs(~ Participant + SentenceType, both)  
both2$frames = 1  
  
subjsum.df = aggregate(cbind(target,frames) ~ win + HeadNoun + SentenceType + Participant + Item, both2, sum)  
subjsum.df$elog <- log( (subjsum.df$target + .5) / (subjsum.df$frames - subjsum.df$target + .5) )  
subjsum.df = subjsum.df[subjsum.df$win>=0 & subjsum.df$win<7,]  
subjsum.df$time=subjsum.df$win\*winsize  
xtabs(~ time,subjsum.df)

## time  
## 0 200 400 600 800 1000 1200   
## 238 238 238 238 238 238 238

# zero looking at DE offset which is time 0  
meansubset = subset(subjsum.df,time == 0)  
meanpartitemdf = aggregate(elog ~ Participant + Item + HeadNoun + SentenceType, meansubset, mean)  
both3 = merge(subjsum.df,meanpartitemdf, all.x=TRUE,by=c("Participant","Item","HeadNoun","SentenceType"),sort=F)  
both3$elog = both3$elog.x-both3$elog.y  
  
# plot figure for dataset  
means.df = aggregate(cbind(elog) ~ time + HeadNoun + SentenceType, both3, mean)  
p = ggplot(means.df , aes( x = time, y = elog, colour=SentenceType))  
p = p + geom\_line()  
p = p + scale\_colour\_brewer(palette="Set1")  
p = p + theme\_bw() # change background to white  
p = p + ylab("Empirical Logit")  
p = p + xlab("Time relative to DE offset")  
#p = p + ylim(0,1)  
p = p + facet\_wrap(~ HeadNoun, ncol=1)  
p+geom\_vline(xintercept = 200\*0:6)



ggsave("elogit.png")

## Saving 5 x 4 in image

Here is a mixed model analysis using 200 ms windows

subjmeans.df = both3  
subjmeans.df$cwin = subjmeans.df$win - mean(subjmeans.df$win)  
subjmeans.df$cCL = ifelse(subjmeans.df$HeadNoun == "DCL",0.5,-0.5)  
subjmeans.df$cobject = ifelse(subjmeans.df$SentenceType == "Object",0.5,-0.5)  
subjmeans.df$Participant=factor(subjmeans.df$Participant)  
subjmeans.df$Item=factor(subjmeans.df$Item)  
  
mixmodel = lmer(elog ~ cwin\*cCL\*cobject + (1 +cobject | Participant) + (1 | Item), subjmeans.df)  
summixed=summary(mixmodel)  
print(summixed)

## Linear mixed model fit by REML ['lmerMod']  
## Formula: elog ~ cwin \* cCL \* cobject + (1 + cobject | Participant) + (1 |   
## Item)  
## Data: subjmeans.df  
##   
## REML criterion at convergence: 7579  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.1401 -0.4911 -0.1439 0.6841 2.1580   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## Participant (Intercept) 0.13472 0.3670   
## cobject 1.17621 1.0845 -0.09  
## Item (Intercept) 0.07545 0.2747   
## Residual 5.26894 2.2954   
## Number of obs: 1666, groups: Participant, 23; Item, 8  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 0.91980 0.13754 6.688  
## cwin 0.23992 0.02822 8.500  
## cCL -0.08615 0.22674 -0.380  
## cobject -0.20484 0.25613 -0.800  
## cwin:cCL -0.05439 0.05645 -0.964  
## cwin:cobject -0.07362 0.05645 -1.304  
## cCL:cobject -0.01836 0.23471 -0.078  
## cwin:cCL:cobject 0.13924 0.11290 1.233  
##   
## Correlation of Fixed Effects:  
## (Intr) cwin cCL cobjct cwn:CL cwn:cb cCL:cb  
## cwin 0.000   
## cCL -0.002 0.000   
## cobject -0.019 0.000 -0.004   
## cwin:cCL 0.000 -0.018 0.000 0.000   
## cwin:cobjct 0.000 0.084 0.000 0.000 -0.011   
## cCL:cobject -0.006 0.000 0.046 -0.002 0.000 0.000   
## cwn:cCL:cbj 0.000 -0.011 0.000 0.000 0.084 -0.018 0.000

print(unique(subjmeans.df$win))

## [1] 0 1 2 3 4 5 6

modellist = modelComparison(mixmodel)

## [1] "remove . ~ . - cwin:cCL:cobject "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + cCL + cobject + (1 + cobject | Participant) + (1 |   
## model2: Item) + cwin:cCL + cwin:cobject + cCL:cobject  
## model: elog ~ cwin \* cCL \* cobject + (1 + cobject | Participant) + (1 |   
## model: Item)  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)  
## model2 12 7583.4 7648.4 -3779.7 7559.4   
## model 13 7583.9 7654.3 -3779.0 7557.9 1.5247 1 0.2169  
## [1] "########## Above comparison for cwin:cCL:cobject "  
## [1] "remove . ~ . - cCL:cobject "

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl =  
## control$checkConv, : Model failed to converge with max|grad| = 0.00200131  
## (tol = 0.002, component 1)

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + cCL + cobject + (1 + cobject | Participant) + (1 |   
## model2: Item) + cwin:cCL + cwin:cobject  
## model: elog ~ cwin + cCL + cobject + (1 + cobject | Participant) + (1 |   
## model: Item) + cwin:cCL + cwin:cobject + cCL:cobject  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)  
## model2 11 7581.4 7641.0 -3779.7 7559.4   
## model 12 7583.4 7648.4 -3779.7 7559.4 0.005 1 0.9434  
## [1] "########## Above comparison for cCL:cobject "  
## [1] "remove . ~ . - cwin:cobject "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + cCL + cobject + (1 + cobject | Participant) + (1 |   
## model2: Item) + cwin:cCL  
## model: elog ~ cwin + cCL + cobject + (1 + cobject | Participant) + (1 |   
## model: Item) + cwin:cCL + cwin:cobject  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)  
## model2 10 7581.1 7635.3 -3780.5 7561.1   
## model 11 7581.4 7641.0 -3779.7 7559.4 1.6449 1 0.1997  
## [1] "########## Above comparison for cwin:cobject "  
## [1] "remove . ~ . - cwin:cCL "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + cCL + cobject + (1 + cobject | Participant) + (1 |   
## model2: Item)  
## model: elog ~ cwin + cCL + cobject + (1 + cobject | Participant) + (1 |   
## model: Item) + cwin:cCL  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)  
## model2 9 7580.3 7629.0 -3781.1 7562.3   
## model 10 7581.1 7635.3 -3780.5 7561.1 1.176 1 0.2782  
## [1] "########## Above comparison for cwin:cCL "  
## [1] "remove . ~ . - cobject "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + cCL + (1 + cobject | Participant) + (1 | Item)  
## model: elog ~ cwin + cCL + cobject + (1 + cobject | Participant) + (1 |   
## model: Item)  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)  
## model2 8 7578.9 7622.2 -3781.4 7562.9   
## model 9 7580.3 7629.0 -3781.1 7562.3 0.6475 1 0.421  
## [1] "########## Above comparison for cobject "  
## [1] "remove . ~ . - cCL "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ cwin + (1 + cobject | Participant) + (1 | Item)  
## model: elog ~ cwin + cCL + (1 + cobject | Participant) + (1 | Item)  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)  
## model2 7 7577.1 7615.0 -3781.5 7563.1   
## model 8 7578.9 7622.2 -3781.4 7562.9 0.1771 1 0.6738  
## [1] "########## Above comparison for cCL "  
## [1] "remove . ~ . - cwin "

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

## Data: subjmeans.df  
## Models:  
## model2: elog ~ (1 + cobject | Participant) + (1 | Item)  
## model: elog ~ cwin + (1 + cobject | Participant) + (1 | Item)  
## Df AIC BIC logLik deviance Chisq Chi Df Pr(>Chisq)   
## model2 6 7648.0 7680.5 -3818.0 7636.0   
## model 7 7577.1 7615.0 -3781.5 7563.1 72.919 1 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## [1] "########## Above comparison for cwin \*\*\* "

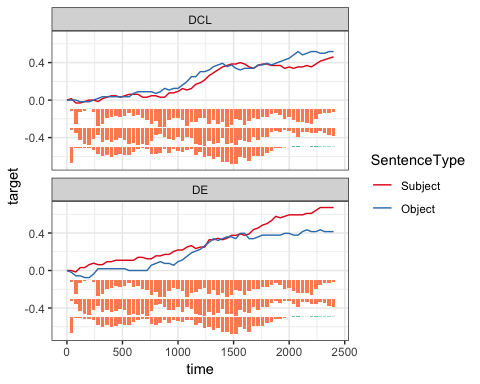
There was a main effect of window, =0.24, SE=0.03, (1)=72.92, p<0.001,

Since there is no three way interaction, we do not report posthocs.

Next we do a permutation analysis.

meansubset = subset(bothdata,time == 0)  
meanpartitemdf = aggregate(target ~ Participant + Item + HeadNoun + SentenceType, meansubset, mean)  
both = merge(bothdata,meanpartitemdf, all.x=TRUE,by=c("Participant","Item","HeadNoun","SentenceType"),sort=F)  
both$target = both$target.x-both$target.y  
  
# create copy of data frame  
pdata = both[!is.na(both$time),]  
wdivsize = 1  
pdata$cobject = ifelse(pdata$SentenceType == "Subject",0.5,-0.5)  
pdata$cDErc = ifelse(pdata$HeadNoun == "DE",0.5,-0.5)  
pdata$win = as.integer(pdata$time/wdivsize)  
#xtabs(~ win, pdata)  
pdata$time = pdata$win\*wdivsize  
  
#pdata$target = pdata$restarg  
# create data frame which averages over subjects.   
# This also stores the results of the permutation analysis  
means.df = aggregate(target ~ SentenceType + cobject + time + HeadNoun, pdata, mean)  
means.df$pstr = 1000  
  
# We do this for each 100 ms window in the data  
timelist = unique(pdata$time)  
for (t in timelist){  
 # create data frame for ONE timebin for each HeadNounuage  
 onetime = subset(pdata,time == t)  
   
 # do regression model on target using structure SentenceTypeition  
 onemodel = summary(lm(target ~ cobject\*cDErc, onetime))  
 # print(summary(onemodel))  
 coefonemodel = coef(onemodel)  
 # this is the t-value for structure  
 objT = coefonemodel[2,3] # observed t-value  
 objP = abs(coefonemodel[2,4]) # observed p-value  
 means.df$objT[means.df$time == t] = objT  
 means.df$objP[means.df$time == t] = objP  
 deT = coefonemodel[3,3] # observed t-value  
 deP = abs(coefonemodel[3,4]) # observed p-value  
 means.df$deT[means.df$time == t] = deT  
 means.df$deP[means.df$time == t] = deP  
 intT = coefonemodel[4,3] # observed t-value  
 intP = abs(coefonemodel[4,4]) # observed p-value  
 means.df$intT[means.df$time == t] = intT  
 means.df$intP[means.df$time == t] = intP  
   
 randDCL = subset(onetime, HeadNoun=="DCL")  
 onemodelDCL = summary(lm(target ~ cobject, randDCL))  
 coefonemodelDCL = coef(onemodelDCL)  
 dclstrT = coefonemodelDCL[2,3] # observed t-value  
 randDE = subset(onetime, HeadNoun=="DE")  
 onemodelDE = summary(lm(target ~ cobject, randDE))  
 coefonemodelDE = coef(onemodelDE)  
 destrT = coefonemodelDE[2,3] # observed t-value  
 means.df$dclstrT[means.df$time == t] = dclstrT  
 means.df$destrT[means.df$time == t] = destrT  
}  
# to see these p-values, we draw them arbitrarily on the graph at 0.2.  
# when the p-value < 0.05, we draw a blue line above 0.2  
# when the p-value > 0.05, we draw an orange line below 0.2  
pliney = -0.1  
plinemax = 0.2  
means.df$plineobjP = pliney+plinemax\*(0.05-means.df$objP)  
means.df$plinecolobjP = ifelse(means.df$objP < 0.05,"a","b")  
plineydeP = -0.3  
means.df$plinedeP = plineydeP+plinemax\*(0.05-means.df$deP)  
means.df$plinecoldeP = ifelse(means.df$deP < 0.05,"a","b")  
plineyInt = -0.5  
means.df$plineintP = plineyInt+plinemax\*(0.05-means.df$intP)  
means.df$plinecolintP = ifelse(means.df$intP < 0.05,"a","b")  
  
  
wsize = mean(pdata$time[2:5] - pdata$time[1:4] - 8)/2  
means.df$SentenceType=factor(means.df$SentenceType,levels=c("Subject","Object"))  
p = ggplot(means.df , aes( x = time, y = target, colour=SentenceType))  
p = p + facet\_wrap(~ HeadNoun, ncol=1)  
p = p + geom\_line()  
p = p + scale\_colour\_brewer(palette="Set1")  
p = p + scale\_fill\_brewer(palette="Set2")  
p = p + theme\_bw() # change background to white  
p = p + geom\_rect(aes(xmin=time-wsize, xmax=time+wsize, ymin = pliney, ymax= plineobjP, fill=plinecolobjP),colour=NA,show.legend=FALSE)  
p = p + geom\_rect(aes(xmin=time-wsize, xmax=time+wsize, ymin = plineydeP, ymax= plinedeP, fill=plinecoldeP),colour=NA,show.legend=FALSE)  
p = p + geom\_rect(aes(xmin=time-wsize, xmax=time+wsize, ymin = plineyInt, ymax= plineintP, fill=plinecolintP),colour=NA,show.legend=FALSE)  
p

## Warning: Removed 4 rows containing missing values (geom\_rect).  
  
## Warning: Removed 4 rows containing missing values (geom\_rect).  
  
## Warning: Removed 4 rows containing missing values (geom\_rect).

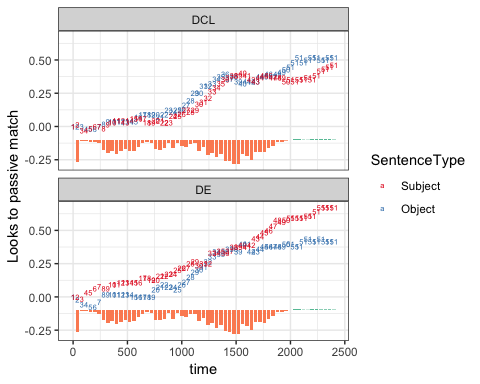


# Also, each window is not independent, so we create clusters for adjacent windows with p<0.05  
# cnum is the cluster number and we increment the number when the p value is > 0.05  
# so clusters with the same cnum are part of the same cluster  
cnum = 1  
lastpval = 100  
lasttdir = 1  
means.df$cnum = 1  
for (t in timelist[2:length(timelist)]){  
 onetime = subset(means.df,time == t & HeadNoun == "DCL" & SentenceType == "Object")  
 pval = abs(onetime$intP)  
 tdir = onetime$intT  
 if (pval < 0.05 & lastpval > 0.05 ){  
 cnum = cnum + 1 # increase cluster number when entering a significant cluster from a non-significant cluster  
 }  
 if (pval > 0.05 ){   
 cnum = cnum + 1 # increase cluster number when not significant  
 }else{  
 # if t value flips direction, even if both are signif,   
 # we should treat those as separate clusters  
 if (lasttdir\*tdir < 0){  
 cnum = cnum + 1   
 }  
 }  
 lastpval = pval  
 lasttdir = tdir  
 means.df$cnum[means.df$time == t] = cnum  
}  
head(means.df,10)

## SentenceType cobject time HeadNoun target pstr objT objP  
## 1 Object -0.5 0 DCL 0.00000000 1000 NaN NaN  
## 2 Subject 0.5 0 DCL 0.00000000 1000 NaN NaN  
## 3 Object -0.5 40 DCL 0.00000000 1000 1.4359235 0.1523596  
## 4 Subject 0.5 40 DCL 0.01538462 1000 1.4359235 0.1523596  
## 5 Object -0.5 80 DCL 0.00000000 1000 0.2501846 0.8026641  
## 6 Subject 0.5 80 DCL -0.03076923 1000 0.2501846 0.8026641  
## 7 Object -0.5 120 DCL -0.01785714 1000 1.2883987 0.1988797  
## 8 Subject 0.5 120 DCL -0.03076923 1000 1.2883987 0.1988797  
## 9 Object -0.5 160 DCL -0.01785714 1000 1.6248256 0.1055459  
## 10 Subject 0.5 160 DCL -0.01538462 1000 1.6248256 0.1055459  
## deT deP intT intP dclstrT destrT  
## 1 NaN NaN NaN NaN NaN NaN  
## 2 NaN NaN NaN NaN NaN NaN  
## 3 -1.4359235 0.1523596 0.1460261 0.88402649 0.92765144 1.099877  
## 4 -1.4359235 0.1523596 0.1460261 0.88402649 0.92765144 1.099877  
## 5 -1.0159650 0.3106954 1.7581830 0.08002373 -1.32226814 1.211492  
## 6 -1.0159650 0.3106954 1.7581830 0.08002373 -1.32226814 1.211492  
## 7 0.4001029 0.6894457 1.7323684 0.08452564 -0.34827214 1.949953  
## 8 0.4001029 0.6894457 1.7323684 0.08452564 -0.34827214 1.949953  
## 9 -0.1633831 0.8703579 1.5512424 0.12219500 0.05241213 2.230050  
## 10 -0.1633831 0.8703579 1.5512424 0.12219500 0.05241213 2.230050  
## plineobjP plinecolobjP plinedeP plinecoldeP plineintP plinecolintP  
## 1 NaN <NA> NaN <NA> NaN <NA>  
## 2 NaN <NA> NaN <NA> NaN <NA>  
## 3 -0.1204719 b -0.3204719 b -0.6668053 b  
## 4 -0.1204719 b -0.3204719 b -0.6668053 b  
## 5 -0.2505328 b -0.3521391 b -0.5060047 b  
## 6 -0.2505328 b -0.3521391 b -0.5060047 b  
## 7 -0.1297759 b -0.4278891 b -0.5069051 b  
## 8 -0.1297759 b -0.4278891 b -0.5069051 b  
## 9 -0.1111092 b -0.4640716 b -0.5144390 b  
## 10 -0.1111092 b -0.4640716 b -0.5144390 b  
## cnum  
## 1 1  
## 2 1  
## 3 2  
## 4 2  
## 5 3  
## 6 3  
## 7 4  
## 8 4  
## 9 5  
## 10 5

plineyInt=-0.1  
means.df$plineintP = plineyInt+plinemax\*(0.05-means.df$intP)  
# this shows the clusters  
p = ggplot(means.df , aes( x = time, y = target, colour=SentenceType, label=cnum))  
p = p + scale\_colour\_brewer(palette="Set1")  
p = p + scale\_fill\_brewer(palette="Set2")  
p = p + theme\_bw() # change background to white  
p = p + ylab("Looks to passive match")  
#p = p + ylim(0,1)  
p = p + facet\_wrap(~ HeadNoun, ncol=1)  
p = p + geom\_rect(aes(xmin=time-wsize, xmax=time+wsize, ymin = plineyInt, ymax= plineintP, fill=plinecolintP),colour=NA,show.legend=FALSE)  
p + geom\_text(size=2)

## Warning: Removed 4 rows containing missing values (geom\_rect).



# we now want to identify the clusters that were significant  
# p-values are same for active and passive, so we just used the active items.  
meansonlyact.df = subset(means.df, HeadNoun == "DCL" & SentenceType == "Object")  
sigcluster = subset(meansonlyact.df, abs(intP) < 0.05 )  
print(sigcluster,digits=3)

## SentenceType cobject time HeadNoun target pstr objT objP deT deP  
## 101 Object -0.5 2000 DCL 0.446 1000 0.713 0.476 1.289 0.199  
## 103 Object -0.5 2040 DCL 0.482 1000 0.489 0.625 1.012 0.313  
## 105 Object -0.5 2080 DCL 0.518 1000 0.351 0.726 0.667 0.505  
## 107 Object -0.5 2120 DCL 0.482 1000 0.337 0.736 1.157 0.249  
## 109 Object -0.5 2160 DCL 0.500 1000 0.294 0.769 1.148 0.252  
## 111 Object -0.5 2200 DCL 0.518 1000 0.200 0.841 1.011 0.313  
## 113 Object -0.5 2240 DCL 0.518 1000 0.610 0.542 1.014 0.312  
## 115 Object -0.5 2280 DCL 0.500 1000 1.014 0.311 1.260 0.209  
## 117 Object -0.5 2320 DCL 0.500 1000 1.241 0.216 1.033 0.302  
## 119 Object -0.5 2360 DCL 0.518 1000 1.224 0.222 0.813 0.417  
## 121 Object -0.5 2400 DCL 0.518 1000 1.325 0.187 0.711 0.478  
## intT intP dclstrT destrT plineobjP plinecolobjP plinedeP plinecoldeP  
## 101 1.97 0.0499 -0.914 1.85 -0.185 b -0.330 b  
## 103 2.42 0.0162 -1.422 1.98 -0.215 b -0.353 b  
## 105 2.55 0.0113 -1.617 1.98 -0.235 b -0.391 b  
## 107 2.05 0.0411 -1.265 1.62 -0.237 b -0.340 b  
## 109 2.02 0.0446 -1.251 1.60 -0.244 b -0.340 b  
## 111 2.37 0.0186 -1.575 1.77 -0.258 b -0.353 b  
## 113 2.37 0.0185 -1.271 2.07 -0.198 b -0.352 b  
## 115 2.13 0.0339 -0.803 2.20 -0.152 b -0.332 b  
## 117 2.16 0.0320 -0.672 2.32 -0.133 b -0.350 b  
## 119 2.17 0.0309 -0.696 2.32 -0.134 b -0.373 b  
## 121 2.07 0.0396 -0.546 2.32 -0.127 b -0.386 b  
## plineintP plinecolintP cnum  
## 101 -0.1000 a 51  
## 103 -0.0932 a 51  
## 105 -0.0923 a 51  
## 107 -0.0982 a 51  
## 109 -0.0989 a 51  
## 111 -0.0937 a 51  
## 113 -0.0937 a 51  
## 115 -0.0968 a 51  
## 117 -0.0964 a 51  
## 119 -0.0962 a 51  
## 121 -0.0979 a 51

# this computes the sum of the t-values for each cluster  
sumcluster = aggregate(cbind(intT,dclstrT,destrT) ~ cnum + HeadNoun, meansonlyact.df, sum)  
head(sumcluster)

## cnum HeadNoun intT dclstrT destrT  
## 1 2 DCL 0.1460261 9.276514e-01 1.099877  
## 2 3 DCL 1.7581830 -1.322268e+00 1.211492  
## 3 4 DCL 1.7323684 -3.482721e-01 1.949953  
## 4 5 DCL 1.5512424 5.241213e-02 2.230050  
## 5 6 DCL 1.5619284 3.564950e-01 2.354545  
## 6 7 DCL 1.3220215 -4.370218e-17 1.754808

# here are the start and finish bits  
timedf = aggregate(time ~ cnum + HeadNoun,sigcluster,min)  
colnames(timedf)<-c("cnum","HeadNoun","starttime")  
timedf2 = aggregate(time ~ cnum + HeadNoun,sigcluster,max)  
timedf$endtime = timedf2$time+33  
print(timedf)

## cnum HeadNoun starttime endtime  
## 1 51 DCL 2000 2433

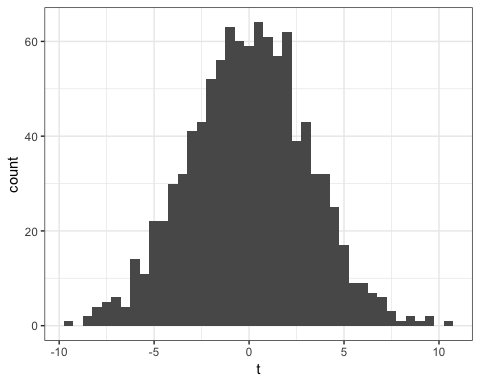
paste(timedf$starttime,"-",timedf$endtime,"ms",sep="",collapse=",")

## [1] "2000-2433ms"

# now we create a distribution of t-values (save in permdist)  
# by randomly scrambling the active and passive labels for each time window 1000 times  
createPermDist <- function(filename="permDist.RData"){  
 n = 1000  
 exptests = data.frame()  
 for (s in 1:length(sigcluster$time)){  
 # print(cl)  
 cl = sigcluster$cnum[s] # cluster number  
 b = sigcluster$time[s] # time  
 print(paste("b ",b))  
 # one time point  
 onetime = subset(pdata, time %in% b)  
 # randSet is a copy of onetime that is scrambled  
 randSet = onetime  
   
 for (i in 1:n){  
 # set.seed(i)  
 # randomly scramble cobject labels without replacement  
 randSet$cobject = sample(randSet$cobject,length(randSet$cobject))  
 randSet$cDErc = sample(randSet$cDErc,length(randSet$cDErc))  
 # test if target is related to random scrambled cobject  
 onemodel = summary(lm(target ~ cobject\*cDErc, randSet))  
 coefonemodel = coef(onemodel)  
 intT = as.numeric(coefonemodel[4,3]) # observed t-value  
 randDCL = subset(randSet, HeadNoun=="DCL")  
 onemodelDCL = summary(lm(target ~ cobject, randDCL))  
 coefonemodelDCL = coef(onemodelDCL)  
 dclstrT = as.numeric(coefonemodelDCL[2,3]) # observed t-value  
 randDE = subset(randSet, HeadNoun=="DE")  
 onemodelDE = summary(lm(target ~ cobject, randDE))  
 coefonemodelDE = coef(onemodelDE)  
 destrT = as.numeric(coefonemodelDE[2,3]) # observed t-value  
   
 df = data.frame(t=intT,cluster=cl,time=b,sim=i,dclT=dclstrT,deT=destrT)  
 exptests = rbind(exptests, df )  
 }  
 }  
 save(exptests,file=filename)  
 return(exptests)  
}  
#exptests = createPermDist("permDistMan5.RData") # since this takes a lot of time, we save the values in a file.  
load("permDistMan5.RData") #this creates data frame exptests from file  
  
# we sum over clusters so that longer clusters have stronger t-values  
sumt.df = aggregate(cbind(t,dclT,deT) ~cluster + sim, exptests, sum)  
head(sumt.df)

## cluster sim t dclT deT  
## 1 51 1 -0.6514369 -5.1960576 1.056916  
## 2 51 2 0.3968350 -0.7606289 2.382795  
## 3 51 3 -4.8081372 0.2780874 -2.904489  
## 4 51 4 -3.7834474 -1.8828483 2.835628  
## 5 51 5 -1.0611246 -2.1225112 5.470188  
## 6 51 6 -3.2347268 2.6891421 -2.965886

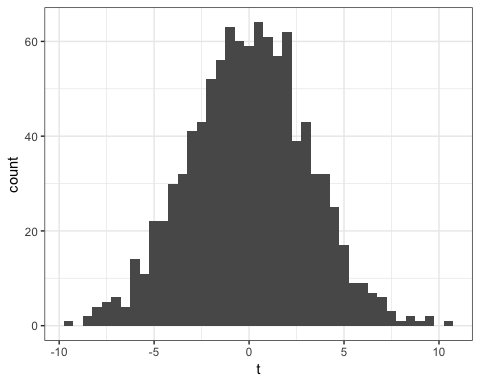
# simulated sum cluster histogram  
p = ggplot(sumt.df,aes(x = t))  
p = p +geom\_histogram(binwidth=0.5)  
p+theme\_bw()



# this code extracts out the maximum sum t for each simulation at each age  
if (length(unique(sumt.df$cluster)) > 1){ # only run if there is more than one cluster  
maxclusterdist = data.frame()  
 for (s in unique(sumt.df$sim)) {  
 # get all results for one simulation in one HeadNounuage  
 onesim = subset(sumt.df,sim == s)  
 onesim$absT = abs(onesim$t)  
 onesim$absdclT = abs(as.numeric(onesim$dclT))  
 onesim$absdeT = abs(as.numeric(onesim$deT))  
 # find max t-value  
 maxrow = onesim[order(onesim$absT,decreasing = T),]  
 maxclusterdist = rbind(maxclusterdist,maxrow[1,])  
 }  
}else{  
 maxclusterdist = sumt.df  
}  
head(maxclusterdist)

## cluster sim t dclT deT  
## 1 51 1 -0.6514369 -5.1960576 1.056916  
## 2 51 2 0.3968350 -0.7606289 2.382795  
## 3 51 3 -4.8081372 0.2780874 -2.904489  
## 4 51 4 -3.7834474 -1.8828483 2.835628  
## 5 51 5 -1.0611246 -2.1225112 5.470188  
## 6 51 6 -3.2347268 2.6891421 -2.965886

# Shows the simulated distribution with maximum cluster t values  
maxclusterdist2 = maxclusterdist[order(maxclusterdist$t),]  
#end = data.frame(xint = maxclusterdist2[c(25,975,1025,1975),]$t)  
p = ggplot(maxclusterdist,aes(x = t))  
p = p +geom\_histogram(binwidth=0.5)  
#p = p +geom\_vline(end,mapping=aes(xintercept=xint))  
p+theme\_bw()

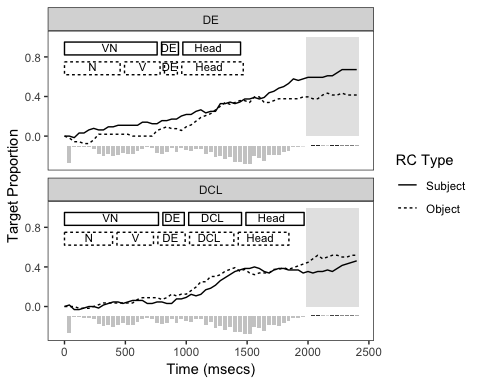


# maxclusterdist is sorted by HeadNounuage,   
# this identifies tvalues that are greater than dist t-values  
for (cl in unique(sumcluster$cnum)){  
 bins = unique(means.df[means.df$cnum == cl,]$time)  
  
 # permtdist is the dist t-values,   
 # so p value is proportion of values greater than observed t-value.  
 # absolute value gives two sided test  
 intT = abs(sumcluster$intT[sumcluster$cnum==cl])  
 intP = sum(abs(maxclusterdist$t) > intT, na.rm = TRUE)/length(maxclusterdist$t)  
 if (intP < 0.05){  
 print(paste("Cluster Interaction Term ",cl,"Obs.sum t",round(intT,3),"PropDist > observed p-value",intP))  
 }  
   
 destrT = abs(sumcluster$destrT[sumcluster$cnum==cl])  
 deT = sum(abs(maxclusterdist$deT) > destrT, na.rm = TRUE)/length(maxclusterdist$deT)  
 if (deT < 0.05){  
 print(paste("Separate Test of deT only",cl,"Obs.sum t",round(destrT,3),"PropDist > observed p-value",deT))  
 }  
   
 dclstrT = abs(sumcluster$dclstrT[sumcluster$cnum==cl])  
 dclT = sum(abs(maxclusterdist$dclT) > dclstrT, na.rm = TRUE)/length(maxclusterdist$dclT)  
 if (dclT < 0.05){  
 print(paste("Separate Test of dclT only",cl,"Obs.sum t",round(dclstrT,3),"PropDist > observed p-value",dclT))  
 }  
  
 means.df$permtestp[means.df$time %in% bins] = intP  
}

## [1] "Cluster Interaction Term 51 Obs.sum t 24.293 PropDist > observed p-value 0"  
## [1] "Separate Test of deT only 51 Obs.sum t 22.011 PropDist > observed p-value 0"  
## [1] "Separate Test of dclT only 51 Obs.sum t 12.03 PropDist > observed p-value 0"

means.df$HeadNoun = factor(means.df$HeadNoun,levels=c("DE","DCL"))  
onsetStim$HeadNoun = factor(onsetStim$HeadNoun,levels=c("DE","DCL"))  
# now we update our plot  
p = ggplot(means.df , aes( x = time, y = target, linetype=SentenceType))  
p = p + facet\_wrap(~ HeadNoun, ncol=1)  
#print(p)  
# this pulls out the clusters which are significant by the permutation test  
meansigStr = subset(means.df,permtestp < 0.025)   
# color them grey  
if (length(meansigStr$time) > 0){  
 p = p + geom\_rect(data=meansigStr,aes(xmin=time-wsize-4, xmax=time+wsize+4, ymin = pliney+0.1, ymax= 1.0),colour=NA,fill="grey90",show.legend=FALSE)  
}  
# same as before  
p = p + geom\_line()  
#p = p + scale\_colour\_brewer(palette="Set1")  
#p = p + scale\_fill\_brewer(palette="Set2")  
p = p + scale\_linetype\_discrete(name="RC Type")  
p = p + theme\_bw() # change background to white  
p = p + ylab("Target Proportion")  
p = p + xlab("Time (msecs)")  
#p = p + ylim(0,1)  
p = p + geom\_rect(aes(xmin=time-wsize, xmax=time+wsize, ymin = plineyInt, ymax= plineintP, fill=plinecolintP),colour=NA,show.legend=FALSE)  
#p = p + geom\_curve(data=meansigPost,aes(x=time+10, xend=time+winsize-10, y = 0.9, yend= 0.9),color="black",size=1, lineend = "square", curvature = -0.5, show.legend=FALSE)  
p = p + geom\_rect(data=onsetStim,aes(xmin=time,xmax = time2-33, ymin=ypos-0.13, ymax=ypos,linetype=SentenceType),fill=NA,colour="black", show.legend=FALSE)  
p = p + geom\_text(data=onsetStim,aes(x=(time+time2)/2, y=ypos-0.05, label=cat), hjust=0.7, size=3,show.legend=FALSE)  
p = p +scale\_colour\_grey()  
p = p +scale\_fill\_grey()  
apaformat(p)

## Warning: Removed 4 rows containing missing values (geom\_rect).



ggsave("permCan.png",width=8,height=8)

## Warning: Removed 4 rows containing missing values (geom\_rect).

accdf = read.csv("ManRC\_accuracy\_data\_all\_final.csv")  
  
accdf$cCL = ifelse(accdf$structure == "DemCL",0.5,-0.5)  
accdf$cobject = ifelse(accdf$extraction == "object",0.5,-0.5)  
accdf$participant=factor(accdf$participant)  
accdf$item = str\_replace(accdf$item,"(S|O)","I")  
accdf$item=factor(accdf$item)  
xtabs( ~ structure + participant,accdf)

## participant  
## structure 1 2 3 4 5 6 7 8 9 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
## bare 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 8  
## DemCL 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 0 0  
## participant  
## structure 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 44 45 46  
## bare 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8  
## DemCL 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

xtabs( ~ extraction + participant,accdf)

## participant  
## extraction 1 2 3 4 5 6 7 8 9 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25  
## object 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## subject 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## participant  
## extraction 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 44 45 46  
## object 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4  
## subject 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4

xtabs( ~ structure + item,accdf)

## item  
## structure I1 I2 I3 I4 I5 I6 I7 I8  
## bare 0 0 0 0 44 44 44 44  
## DemCL 44 44 44 44 0 0 0 0

xtabs( ~ extraction + item,accdf)

## item  
## extraction I1 I2 I3 I4 I5 I6 I7 I8  
## object 22 22 22 22 22 22 22 22  
## subject 22 22 22 22 22 22 22 22

acc.glm = glmer(Correct ~ cobject\*cCL + (1 + extraction | participant) + (1 | item),accdf,family="binomial")

## boundary (singular) fit: see ?isSingular

print(summary(acc.glm))

## Generalized linear mixed model fit by maximum likelihood (Laplace  
## Approximation) [glmerMod]  
## Family: binomial ( logit )  
## Formula: Correct ~ cobject \* cCL + (1 + extraction | participant) + (1 |   
## item)  
## Data: accdf  
##   
## AIC BIC logLik deviance df.resid   
## 450.8 481.7 -217.4 434.8 344   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.3237 -1.0366 0.5140 0.6778 1.0806   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr   
## participant (Intercept) 0.61285 0.7828   
## extractionsubject 2.30679 1.5188 -1.00  
## item (Intercept) 0.01764 0.1328   
## Number of obs: 352, groups: participant, 44; item, 8  
##   
## Fixed effects:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) 0.7868 0.1360 5.786 7.21e-09 \*\*\*  
## cobject -0.5128 0.3415 -1.501 0.133   
## cCL 0.1445 0.2616 0.552 0.581   
## cobject:cCL -0.2394 0.6664 -0.359 0.719   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) cobjct cCL   
## cobject -0.060   
## cCL 0.051 -0.024   
## cobject:cCL -0.022 0.027 -0.027  
## convergence code: 0  
## boundary (singular) fit: see ?isSingular

accdf$pred = remef(acc.glm, ran = "all")  
head(accdf)

## item participant extraction Correct structure cCL cobject pred  
## 1 I1 1 subject 0 DemCL 0.5 -0.5 -0.4123535  
## 2 I2 1 subject 1 DemCL 0.5 -0.5 1.9350090  
## 3 I3 1 subject 1 DemCL 0.5 -0.5 1.9657572  
## 4 I4 1 subject 1 DemCL 0.5 -0.5 1.9554244  
## 5 I1 1 object 1 DemCL 0.5 0.5 1.4737056  
## 6 I2 1 object 1 DemCL 0.5 0.5 1.4115142

meandf = aggregate(Correct ~ extraction + structure, accdf, mean)  
print(meandf)

## extraction structure Correct  
## 1 object bare 0.6136364  
## 2 subject bare 0.6931818  
## 3 object DemCL 0.6136364  
## 4 subject DemCL 0.7386364

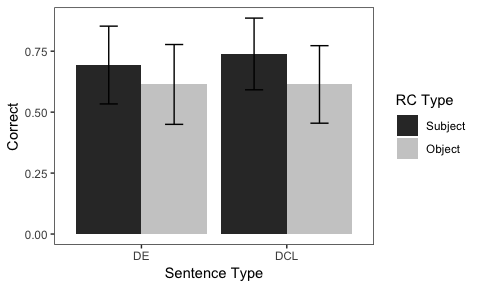
# compute sd from pred without random effects  
meandf$sd = aggregate(pred ~ extraction + structure, accdf,sd)$pred  
print(meandf)

## extraction structure Correct sd  
## 1 object bare 0.6136364 1.0861246  
## 2 subject bare 0.6931818 1.0582441  
## 3 object DemCL 0.6136364 1.0546698  
## 4 subject DemCL 0.7386364 0.9755494

# se is computed from sd divided by number of participants  
meandf$se = meandf$sd/sqrt( nlevels(accdf$participant) )  
meandf$upper = meandf$Correct + meandf$se  
meandf$lower = meandf$Correct - meandf$se  
print(meandf)

## extraction structure Correct sd se upper lower  
## 1 object bare 0.6136364 1.0861246 0.1637394 0.7773758 0.4498969  
## 2 subject bare 0.6931818 1.0582441 0.1595363 0.8527181 0.5336455  
## 3 object DemCL 0.6136364 1.0546698 0.1589975 0.7726338 0.4546389  
## 4 subject DemCL 0.7386364 0.9755494 0.1470696 0.8857060 0.5915668

meandf$extraction=factor(meandf$extraction,labels=c("Subject","Object"),levels=c("subject","object"))  
meandf$structure=factor(meandf$structure,labels=c("DE","DCL"))  
  
p = ggplot(meandf, aes(x=structure,y=Correct, fill=extraction,ymin = lower, ymax=upper))  
p = p +geom\_bar(stat="identity",position="dodge")  
p = p +scale\_fill\_grey(name="RC Type")  
p = p + xlab("Sentence Type")  
p = p + geom\_errorbar(width=0.25, position=position\_dodge(.9) )  
apaformat(p)



ggsave("accuracy.png",width=5,height=3)