

Silent and spoken intervals in naming: Manuscript workflow

Athanassios Protopapas

9 February 2023

Contents

1	Introduction	2
2	Graphs of speech and silent intervals	2
2.1	Descriptives tables	4
3	Comparisons among mean times	7
3.1	Comparisons of discrete articulation duration among tasks	7
3.2	Comparison of serial articulation time across tasks	9
3.3	Articulation duration: Interactions between task and format	10
3.4	Articulation duration: Simple comparison of formats by task and grade	11
4	Correlations between discrete and serial	12
5	Relationship between silent and spoken intervals in discrete tasks	14
6	Association of serial advantage with discrete articulation	15
6.1	Graphical comparison	15
6.2	Statistical comparison of grade medians	17
6.3	Serial advantage vs. articulation: individual differences (all children)	18
7	Association of total serial time with discrete onset latency	20
7.1	Graph of serial total time vs. discrete onset latency	20
7.2	Correlations between serial pause+articulation and discrete onset latency	21
7.3	Statistical comparison between serial pause+articulation and discrete onset latency	23
8	Associations of naming response components with reading measures	25
8.1	Descriptives for reading measures	25
8.2	Tables of correlations	25
8.3	Graphical display of correlations	27
9	Session information	29

```
library(psych)
require(nlme)
require(multcomp)
require(emmeans)
library(signs)
```

1 Introduction

In this document we present analyses of silent and spoken intervals in discrete and serial naming tasks as they are presented in the submitted manuscript.

In separate supplementary documents we present:

- The data checking procedures for each type of data, including scaling considerations, distributions, and intercorrelations, as well information about the frequency and extent of each type of event considered in the serial naming tasks (S1_check_data).
- Definition and description of each variable and basic descriptive analyses, including graphical presentation of individual-level data underlying the group comparisons and intercorrelations among measures from related sets of tasks (S2_descriptives).
- Basic exploratory comparisons and supplementary analyses of the serial and discrete naming data (S3_main_analyses).
- Computations and analyses focusing on the serial advantage and the serial articulation penalty (S4_seradv).

```
read.table("read_discrete_serial_data.txt",stringsAsFactors=TRUE) -> rsdw
tasks <- c("digit","numword","word","dice","object")
tcols <- c("#D55E00","#E69F00","#009E73","#0180C7","#884584")
```

2 Graphs of speech and silent intervals

The following graphs display mean durations of spoken and silent intervals per item in discrete and serial naming tasks (computed as the geometric means over participants) for each task and grade.

```
#pdf("serdis_silspk.pdf",width=10,height=10)
par(mfcol=c(2,2))
par(oma=c(2,4,2,2))
par(mai=c(0.05,0.05,0.05,0.05))
par(mgp=c(2,0.6,0))
par(tcl= -0.2)

yl <- c(0,1.1)

discons <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logonsRT",tasks,sep="."),collapse="),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=yl,xaxt="n",xlab="",ylab="",las=1,xpd=NA,cex.axis=1.5)
axis(1,at=1:3,labels=NA,cex=1.5)
for (itsk in 1:length(tasks)) {
  tsk <- paste("logonsRT",tasks[itsk],sep=".")
  lines(1:3,discons[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,discons[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}
#legend(2.5,1,tasks,col=tcols,pch=21,lty=1,lwd=3)
mtext("Discrete",3,cex=1.75,line=0.70)

discart <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logartic",tasks,sep="."),collapse="),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=yl,xaxt="n",xlab="",ylab="",las=1,xpd=NA,cex.axis=1.5)
axis(1,at=1:3,labels=paste("Grade",c(1,3,5),sep=" "),cex.axis=1.5)
for (itsk in 1:length(tasks)) {
  tsk <- paste("logartic",tasks[itsk],sep=".")
```

```

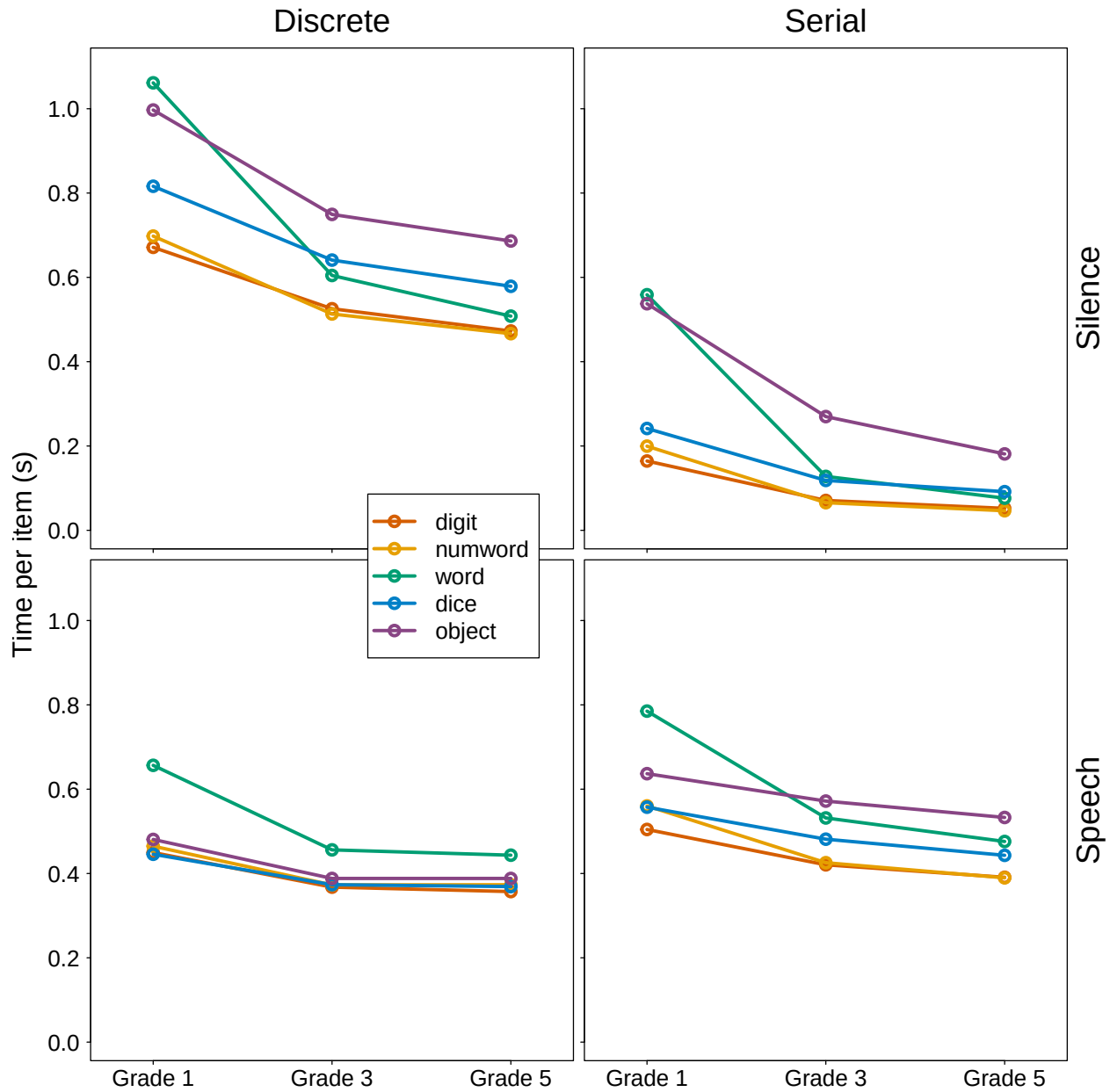
    lines(1:3,discart[,tsk],col=tcols[itsk],lwd=3)
    points(1:3,discart[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
  }
  legend(2.20,1.30,tasks,col=tcols,pch=21,lty=1,lwd=3,cex=1.5,xpd=NA,bg="white")

serpau <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logpautE",tasks,sep="."),collapse=","),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=yl,xaxt="n",yaxt="n",xlab="",ylab="",las=1)
axis(1,at=1:3,labels=NA,cex=1.5)
axis(2,labels=NA)
for (itsk in 1:length(tasks)) {
  tsk <- paste("logpautE",tasks[itsk],sep=".")
  lines(1:3,serpau[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,serpau[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}
#legend(1.80,0.95,tasks,col=tcols,pch=21,lty=1,lwd=3,cex=1.5)
mtext("Serial",3,cex=1.75,line=0.70)
mtext("Silence",4,cex=1.75,line=0.95)

serart <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logarttE",tasks,sep="."),collapse=","),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=yl,xaxt="n",yaxt="n",xlab="",ylab="",las=1)
axis(1,at=1:3,labels=paste("Grade",c(1,3,5),sep=" "),cex.axis=1.5)
axis(2,labels=NA)
for (itsk in 1:length(tasks)) {
  tsk <- paste("logarttE",tasks[itsk],sep=".")
  lines(1:3,serart[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,serart[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}
#legend(2.5,0.8,tasks,col=tcols,pch=21,lty=1,lwd=3)
mtext("Speech",4,cex=1.75,line=0.95)

mtext("Time per item (s)",side=2,outer=T,cex=1.5,line=2.75)

```



#dev.off()

This is Figure 2 of the manuscript.

2.1 Descriptives tables

The following table lists descriptive statistics for the data plotted in the above graph, back-transformed to time before calculating the statistics. These descriptives are not justified by the underlying distributions and are not used in any other displays of statistical comparisons, they are only listed for illustration.

```
disconst <- cbind(
  as.vector(t(aggregate(
    as.formula( paste("cbind(",
                      paste(paste("exp(logonsRT",tasks,sep="."),
                                collapse="),"), "))) ~ grade",
```

```

        sep="" ) ,
    rsdw, drop=FALSE, na.action=na.pass, FUN=mean, na.rm=T )[, -1]),
as.vector(t(aggregate(
  as.formula( paste("cbind(",
    paste(paste("exp(logonsRT", tasks, sep="."),
      collapse="),"), ") ~ grade",
    sep="" ) ,
    rsdw, drop=FALSE, na.action=na.pass, FUN=sd, na.rm=T )[, -1]),
as.vector(t(aggregate(
  as.formula( paste("cbind(",
    paste(paste("exp(logonsRT", tasks, sep="."),
      collapse="),"), ") ~ grade",
    sep="" ) ,
    rsdw, drop=FALSE, na.action=na.pass, FUN=min, na.rm=T )[, -1]),
as.vector(t(aggregate(
  as.formula( paste("cbind(",
    paste(paste("exp(logonsRT", tasks, sep="."),
      collapse="),"), ") ~ grade",
    sep="" ) ,
    rsdw, drop=FALSE, na.action=na.pass, FUN=max, na.rm=T )[, -1]))
)
disconst <- round( cbind( disconst[1:5,], disconst[6:10,], disconst[11:15,] ), 0 )
colnames(disconst) <- paste( rep(paste("G", c(1,3,5), sep=""), each=4),
  rep(c("M", "SD", "min", "max"), 3),
  sep="." )
rownames(disconst) <- tasks

discartt <- cbind(
  as.vector(t(aggregate(
    as.formula( paste("cbind(",
      paste(paste("exp(logartic", tasks, sep="."),
        collapse="),"), ") ~ grade",
        sep="" ) ,
    rsdw, drop=FALSE, na.action=na.pass, FUN=mean, na.rm=T )[, -1]),
as.vector(t(aggregate(
  as.formula( paste("cbind(",
    paste(paste("exp(logartic", tasks, sep="."),
      collapse="),"), ") ~ grade",
      sep="" ) ,
    rsdw, drop=FALSE, na.action=na.pass, FUN=sd, na.rm=T )[, -1]),
as.vector(t(aggregate(
  as.formula( paste("cbind(",
    paste(paste("exp(logartic", tasks, sep="."),
      collapse="),"), ") ~ grade",
      sep="" ) ,
    rsdw, drop=FALSE, na.action=na.pass, FUN=min, na.rm=T )[, -1]),
as.vector(t(aggregate(
  as.formula( paste("cbind(",
    paste(paste("exp(logartic", tasks, sep="."),
      collapse="),"), ") ~ grade",
      sep="" ) ,
    rsdw, drop=FALSE, na.action=na.pass, FUN=max, na.rm=T )[, -1]))
)

```

```

discartt <- round( cbind( discartt[1:5,], discartt[6:10,], discartt[11:15,] ), 0 )
colnames(discartt) <- paste( rep(paste("G",c(1,3,5),sep=""),each=4),
                             rep(c("M","SD","min","max"),3),
                             sep="." )
rownames(discartt) <- tasks

serpaut <- cbind(
  as.vector(t(aggregate(
    as.formula( paste("cbind(",
                      paste(paste("exp(logpautE",tasks,sep="."),
                                collapse="),"), "))) ~ grade",
                      sep="") ),
    rsdw, drop=FALSE, na.action=na.pass, FUN=mean, na.rm=T )[, -1])),
  as.vector(t(aggregate(
    as.formula( paste("cbind(",
                      paste(paste("exp(logpautE",tasks,sep="."),
                                collapse="),"), "))) ~ grade",
                      sep="") ),
    rsdw, drop=FALSE, na.action=na.pass, FUN=sd, na.rm=T )[, -1])),
  as.vector(t(aggregate(
    as.formula( paste("cbind(",
                      paste(paste("exp(logpautE",tasks,sep="."),
                                collapse="),"), "))) ~ grade",
                      sep="") ),
    rsdw, drop=FALSE, na.action=na.pass, FUN=min, na.rm=T )[, -1])),
  as.vector(t(aggregate(
    as.formula( paste("cbind(",
                      paste(paste("exp(logpautE",tasks,sep="."),
                                collapse="),"), "))) ~ grade",
                      sep="") ),
    rsdw, drop=FALSE, na.action=na.pass, FUN=max, na.rm=T )[, -1]))
)
serpaut <- round( 1000.0 * cbind( serpaut[1:5,], serpaut[6:10,], serpaut[11:15,] ), 0 )
colnames(serpaut) <- paste( rep(paste("G",c(1,3,5),sep=""),each=4),
                             rep(c("M","SD","min","max"),3),
                             sep="." )
rownames(serpaut) <- tasks

serartt <- cbind(
  as.vector(t(aggregate(
    as.formula( paste("cbind(",
                      paste(paste("exp(logarttE",tasks,sep="."),
                                collapse="),"), "))) ~ grade",
                      sep="") ),
    rsdw, drop=FALSE, na.action=na.pass, FUN=mean, na.rm=T )[, -1])),
  as.vector(t(aggregate(
    as.formula( paste("cbind(",
                      paste(paste("exp(logarttE",tasks,sep="."),
                                collapse="),"), "))) ~ grade",
                      sep="") ),
    rsdw, drop=FALSE, na.action=na.pass, FUN=sd, na.rm=T )[, -1])),

```

```

as.vector(t(aggregate(
  as.formula( paste("cbind(",
    paste(paste("exp(logarttE",tasks,sep="."),
      collapse="),"), "))) ~ grade",
    sep="") ),
  rsdw, drop=FALSE, na.action=na.pass, FUN=min, na.rm=T )[, -1]))
as.vector(t(aggregate(
  as.formula( paste("cbind(",
    paste(paste("exp(logarttE",tasks,sep="."),
      collapse="),"), "))) ~ grade",
    sep="") ),
  rsdw, drop=FALSE, na.action=na.pass, FUN=max, na.rm=T )[, -1]))
)
serartt <- round( 1000.0 * cbind( serartt[1:5,], serartt[6:10,], serartt[11:15,] ), 0 )
colnames(serartt) <- paste( rep(paste("G",c(1,3,5),sep=""),each=4),
  rep(c("M","SD","min","max"),3),
  sep="." )
rownames(serartt) <- tasks

rbind(disconst, discartt, serpaut, serartt)

```

##	G1.M	G1.SD	G1.min	G1.max	G3.M	G3.SD	G3.min	G3.max	G5.M	G5.SD	G5.min	G5.max
## digit	684	130	482	1179	531	82	308	817	476	63	368	677
## numword	722	164	489	1548	519	76	320	712	470	63	365	669
## word	1110	350	600	2266	622	167	367	1356	513	78	383	816
## dice	845	198	550	1594	653	129	370	1180	584	78	422	773
## object	1013	165	728	1573	759	130	483	1227	695	125	476	1082
## digit	464	120	231	983	378	89	214	660	369	97	195	677
## numword	484	136	248	901	383	94	233	766	385	99	193	703
## word	682	203	374	1580	467	106	264	944	455	109	230	832
## dice	462	128	246	872	384	91	225	655	380	95	214	707
## object	496	128	245	887	402	99	212	806	398	93	203	706
## digit	196	108	45	515	88	66	15	427	70	64	11	346
## numword	237	146	39	777	82	68	12	453	62	56	6	317
## word	704	476	125	2185	171	158	22	852	99	77	12	480
## dice	278	153	51	824	140	91	28	580	117	93	11	573
## object	587	247	190	1548	296	138	94	742	202	106	50	521
## digit	512	75	362	688	421	53	312	570	394	62	281	613
## numword	568	94	383	839	425	55	326	598	393	59	300	578
## word	810	191	486	1258	541	107	346	862	481	80	334	733
## dice	564	93	387	860	483	69	359	690	446	61	304	611
## object	646	95	412	910	575	71	375	744	537	70	407	735

This is the content of Table 1 in the manuscript.

3 Comparisons among mean times

Interestingly, articulation times seem to differ among tasks despite the psycholinguistic matching. We next test which of these points differ significantly.

3.1 Comparisons of discrete articulation duration among tasks

In the following set of post-hoc contrasts we start with adjacent points (within each grade) and add comparisons of points in increasing distance until a difference is established.

```

artnm <- paste("logartic",tasks,sep=".")
artlong <- reshape(rsdw[,c("sID", "grade", artnm)], direction="long",
  varying=artnm, timevar="task", times=tasks, idvar="sID", v.names="logartic")
artlong$task <- relevel(as.factor(artlong$task), ref="word")
artlong$grade <- relevel(artlong$grade, ref="G3")
artlong <- droplevels(artlong[!is.na(artlong$logartic),])
l0 <- lme(logartic~task*grade, random=~1|sID/task, data=artlong)
contrast.matrix.tasks <- rbind(
  #
  # order G1: digit - numword - object - word
  "G1 digit-digit" = c( 0, +1, -1, 0, 0, 0, 0, +1, -1, 0, 0, 0, 0, 0, 0, 0 ),
  "G1 digit-numword" = c( 0, +1, 0, -1, 0, 0, 0, +1, 0, -1, 0, 0, 0, 0, 0, 0 ),
  "G1 digit-object" = c( 0, 0, +1, -1, 0, 0, 0, 0, +1, -1, 0, 0, 0, 0, 0, 0 ),
  "G1 digit-word" = c( 0, 0, 0, +1, 0, 0, 0, 0, +1, 0, -1, 0, 0, 0, 0, 0 ),
  "G1 numword-object" = c( 0, 0, 0, +1, -1, 0, 0, 0, 0, +1, -1, 0, 0, 0, 0, 0 ),
  "G1 dice-object" = c( 0, +1, 0, 0, -1, 0, 0, 0, +1, 0, 0, -1, 0, 0, 0, 0 ),
  "G1 object-word" = c( 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0 ),
  # order G3: digit - numword - dice - object - word
  "G3 digit-numword" = c( 0, 0, +1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 digit-dice" = c( 0, -1, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 digit-object" = c( 0, 0, 0, +1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 numword-dice" = c( 0, -1, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 numword-object" = c( 0, 0, 0, 0, +1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 dice-object" = c( 0, +1, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 object-word" = c( 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  # order G5: digit - dice - numword - object - word
  "G5 digit-dice" = c( 0, -1, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, +1, 0, 0 ),
  "G5 digit-numword" = c( 0, 0, +1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, -1 ),
  "G5 digit-object" = c( 0, 0, 0, +1, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0 ),
  "G5 dice-numword" = c( 0, +1, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0 ),
  "G5 dice-object" = c( 0, +1, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1 ),
  "G5 numword-object" = c( 0, 0, 0, 0, +1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1 ),
  "G5 numword-word" = c( 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G5 object-word" = c( 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0 )
)
summary(glht(l0, contrast.matrix.tasks))

```

```

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lme.formula(fixed = logartic ~ task * grade, data = artlong,
## random = ~1 | sID/task)
##
## Linear Hypotheses:
##
## Estimate Std. Error z value Pr(>|z|)
## G1 digit-digit == 0 -0.007374 0.015085 -0.489 1.00000
## G1 digit-numword == 0 -0.045303 0.015136 -2.993 0.05239 .
## G1 digit-object == 0 -0.037929 0.015136 -2.506 0.19785
## G1 digit-word == 0 -0.067985 0.015085 -4.507 < 0.001 ***
## G1 numword-object == 0 -0.030056 0.015136 -1.986 0.53939
## G1 dice-object == 0 -0.075359 0.015085 -4.995 < 0.001 ***
## G1 object-word == 0 -0.312421 0.015233 -20.509 < 0.001 ***
## G3 digit-numword == 0 -0.014305 0.014953 -0.957 0.99584
## G3 digit-dice == 0 -0.017340 0.014907 -1.163 0.97849

```

```
## G3 digit-object == 0 -0.060449 0.014907 -4.055 0.00107 **
## G3 numword-dice == 0 -0.003035 0.014907 -0.204 1.00000
## G3 numword-object == 0 -0.046144 0.014907 -3.095 0.03784 *
## G3 dice-object == 0 -0.043109 0.014862 -2.901 0.06864 .
## G3 object-word == 0 -0.154864 0.014862 -10.420 < 0.001 ***
## G5 digit-dice == 0 -0.032673 0.015289 -2.137 0.42331
## G5 digit-numword == 0 -0.044176 0.015240 -2.899 0.06940 .
## G5 digit-object == 0 -0.084185 0.015240 -5.524 < 0.001 ***
## G5 dice-numword == 0 -0.011503 0.015289 -0.752 0.99958
## G5 dice-object == 0 -0.051512 0.015289 -3.369 0.01495 *
## G5 numword-object == 0 -0.040009 0.015240 -2.625 0.14720
## G5 numword-word == 0 -0.171702 0.015240 -11.266 < 0.001 ***
## G5 object-word == 0 -0.131693 0.015240 -8.641 < 0.001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

In every grade, articulation duration was longer for words than for every other task, and longer for objects than for digits and dice. Articulation duration did not differ in any grade between dice, digit, and number words, in all of which the exact same words were the responses, equally repeated.

3.2 Comparison of serial articulation time across tasks

The pattern of means appears qualitatively different for serial naming, compared to discrete naming. The following set of post-hoc contrasts tests which of these points differ significantly for each grade.

```
serartnm <- paste("logarttE",tasks,sep=".")
serartlong <- reshape(rsdw[,c("sID", "grade", serartnm)], direction="long",
                      varying=serartnm, timevar="task", times=tasks, idvar="sID", v.names="logarttE")
serartlong$task <- relevel(as.factor(serartlong$task), ref="word")
serartlong$grade <- relevel(serartlong$grade, ref="G3")
serartlong <- droplevels(serartlong[!is.na(serartlong$logarttE),])
l2 <- lme(logarttE~task*grade, random=~1|sID/task, data=serartlong)
#anova(l2)
#print(summary(l2))
contrast.matrix.tasks <- rbind(
  #
  "G1 digit-numword" = c( 0, 0, +1, -1, 0, 0, 0, 0, 0, +1, -1, 0, 0, 0, 0, 0 ),
  "G1 numword-dice" = c( 0, -1, 0, +1, 0, 0, 0, 0, -1, 0, +1, 0, 0, 0, 0, 0 ),
  "G1 digit-dice" = c( 0, -1, +1, 0, 0, 0, 0, 0, -1, +1, 0, 0, 0, 0, 0, 0 ),
  "G1 dice-object" = c( 0, +1, 0, 0, -1, 0, 0, 0, +1, 0, 0, -1, 0, 0, 0, 0 ),
  "G1 object-word" = c( 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0 ),

  "G3 digit-numword" = c( 0, 0, +1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 numword-dice" = c( 0, -1, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 digit-dice" = c( 0, -1, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 dice-word" = c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),
  "G3 word-object" = c( 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ),

  "G5 digit-numword" = c( 0, 0, +1, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, -1, 0 ),
  "G5 numword-dice" = c( 0, -1, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, -1, 0, +1, 0 ),
  "G5 digit-dice" = c( 0, -1, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1, +1, 0, 0 ),
  "G5 dice-word" = c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0 ),
  "G5 word-object" = c( 0, 0, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, -1 )
)
```

```
summary(glht(l2,contrast.matrix.tasks))
```

```
##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lme.formula(fixed = logarttE ~ task * grade, data = serartlong,
## random = ~1 | sID/task)
##
## Linear Hypotheses:
## Estimate Std. Error z value Pr(>|z|)
## G1 digit-numword == 0 -0.1017177 0.0138909 -7.323 < 1e-06 ***
## G1 numword-dice == 0 0.0035150 0.0138272 0.254 1.000
## G1 digit-dice == 0 -0.0982027 0.0139343 -7.048 < 1e-06 ***
## G1 dice-object == 0 -0.1347041 0.0138468 -9.728 < 1e-06 ***
## G1 object-word == 0 -0.2090768 0.0137588 -15.196 < 1e-06 ***
## G3 digit-numword == 0 -0.0097313 0.0135517 -0.718 0.999
## G3 numword-dice == 0 -0.1247522 0.0136350 -9.149 < 1e-06 ***
## G3 digit-dice == 0 -0.1344835 0.0136757 -9.834 < 1e-06 ***
## G3 dice-word == 0 -0.1040112 0.0136757 -7.606 < 1e-06 ***
## G3 word-object == 0 -0.0711783 0.0136337 -5.221 2.52e-06 ***
## G5 digit-numword == 0 0.0008077 0.0138550 0.058 1.000
## G5 numword-dice == 0 -0.1278412 0.0138550 -9.227 < 1e-06 ***
## G5 digit-dice == 0 -0.1270335 0.0138550 -9.169 < 1e-06 ***
## G5 dice-word == 0 -0.0706516 0.0139433 -5.067 5.54e-06 ***
## G5 word-object == 0 -0.1144809 0.0141709 -8.079 < 1e-06 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

Indeed the results confirm that this pattern is nothing like the pattern for discrete articulation times. All points are significantly different except for three nearly identical pairs, namely dice and number words in Grade 1 and digits and number words in Grades 3 and 5.

3.3 Articulation duration: Interactions between task and format

Articulation durations differ among serial tasks in most cases. In contrast, articulation durations did not differ for discrete dice, digits, and number words. To directly examine whether the different patterns of significance are statistically significant, we test the interaction between task and format among task triplets: The aforementioned three tasks, which are of special importance due to the identical responses, and (separately) the three more difficult tasks, namely dice, objects, and words. (Note that dice is included in both triplets.)

```
artlong$format <- "discrete"
serartlong$format <- "serial"
names(serartlong)[names(serartlong)=="logarttE"] <- "logartic"
artboth <- rbind(artlong,serartlong)
artboth$format <- as.factor(artboth$format)
artboth$logartic[artboth$format=="discrete"] <- log(exp(artboth$logartic[artboth$format=="discrete"])/1)

li <- lme(logartic~task*format*grade,random=~1|sID/task/format,data=artboth)

# use emmeans to derive the interaction contrast matrix, which is more complicated than the direct comp
emmeans(li,pairwise=task*format|grade) -> apbg # all pairs by grade
contrast(apbg[[1]],interaction=c(task="pairwise",format="pairwise"),by="grade") -> cint
concoef <- as.list(cint)$linfct
```

```

conname <- expand.grid(gsub(" ", "", levels(summary(cint)$task_pairwise)), levels(summary(cint)$grade))

# then select the rows of interest (concerning the three tasks with identical responses)
conset <- c(15,16,18, 11,14,17, 5,6,8, 1,4,7, 25,26,28, 21,24,27)
contrast.matrix.interactions <- concoef[conset,]
rownames(contrast.matrix.interactions) <- apply(conname[conset,], 1, function(x){paste(x[2], x[1], sep=" ")})

summary(glht(li, contrast.matrix.interactions))

```

```

##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lme.formula(fixed = logartic ~ task * format * grade, data = artboth,
## random = ~1 | sID/task/format)
##
## Linear Hypotheses:
##
## Estimate Std. Error z value Pr(>|z|)
## G1 dice-digit == 0 -0.109347 0.027714 -3.945 0.00139 **
## G1 dice-numword == 0 -0.043226 0.027665 -1.562 0.81207
## G1 digit-numword == 0 0.066120 0.027714 2.386 0.22911
## G1 word-dice == 0 0.047560 0.027713 1.716 0.70641
## G1 word-object == 0 0.104403 0.027674 3.773 0.00272 **
## G1 dice-object == 0 0.056844 0.027635 2.057 0.44104
## G3 dice-digit == 0 -0.116777 0.027305 -4.277 < 0.001 ***
## G3 dice-numword == 0 -0.119058 0.027259 -4.368 < 0.001 ***
## G3 digit-numword == 0 -0.002281 0.027221 -0.084 1.00000
## G3 word-dice == 0 0.093912 0.027260 3.445 0.00972 **
## G3 word-object == 0 0.227743 0.027221 8.366 < 0.001 ***
## G3 dice-object == 0 0.133830 0.027298 4.903 < 0.001 ***
## G5 dice-digit == 0 -0.093906 0.027839 -3.373 0.01255 *
## G5 dice-numword == 0 -0.138889 0.027839 -4.989 < 0.001 ***
## G5 digit-numword == 0 -0.044984 0.027799 -1.618 0.77598
## G5 word-dice == 0 0.110559 0.027919 3.960 0.00128 **
## G5 word-object == 0 0.242142 0.028088 8.621 < 0.001 ***
## G5 dice-object == 0 0.131583 0.028046 4.692 < 0.001 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)

```

The results confirm that the difference between articulation duration for dice and digits is significantly larger in serial than in discrete tasks in every grade; and so is the difference between dice and number words in Grades 3 and 5. Most other differences are also significantly larger in serial than discrete format.

3.4 Articulation duration: Simple comparison of formats by task and grade

Finally we can directly compare the duration of spoken intervals for each task and grade based on the same multivariate model fitted above.

```

contrast(apbg[[1]], method="pairwise", simple="format", by=c("grade", "task")) -> cserdis
concoef <- as.list(cserdis)$linfct
conname <- expand.grid(levels(summary(cserdis)$task), levels(summary(cserdis)$grade))
rownames(concoef) <- apply(conname, 1, function(x){paste(x[2], x[1], sep=" ")})
contrast.matrix.simple <- concoef
summary(glht(li, contrast.matrix.simple))

```

```
##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lme.formula(fixed = logartic ~ task * format * grade, data = artboth,
## random = ~1 | sID/task/format)
##
## Linear Hypotheses:
##           Estimate Std. Error z value Pr(>|z|)
## G3 word == 0    -0.14875    0.01922  -7.739 < 1e-10 ***
## G3 dice == 0     -0.24266    0.01933 -12.554 < 1e-10 ***
## G3 digit == 0    -0.12589    0.01927  -6.531 9.78e-10 ***
## G3 numword == 0  -0.12360    0.01922  -6.430 1.91e-09 ***
## G3 object == 0   -0.37649    0.01927 -19.533 < 1e-10 ***
## G1 word == 0     -0.18284    0.01962  -9.317 < 1e-10 ***
## G1 dice == 0     -0.23040    0.01957 -11.774 < 1e-10 ***
## G1 digit == 0    -0.12105    0.01962  -6.168 1.04e-08 ***
## G1 numword == 0  -0.18717    0.01957  -9.565 < 1e-10 ***
## G1 object == 0   -0.28724    0.01951 -14.721 < 1e-10 ***
## G5 word == 0     -0.07219    0.01977  -3.651 0.00391 **
## G5 dice == 0     -0.18275    0.01971  -9.270 < 1e-10 ***
## G5 digit == 0    -0.08884    0.01966  -4.520 9.29e-05 ***
## G5 numword == 0  -0.04386    0.01966  -2.231 0.32301
## G5 object == 0   -0.31433    0.01995 -15.756 < 1e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

These comparisons confirm that spoken interval durations are significantly longer for serial than for the corresponding discrete tasks in every case except number words in Grade 5.

4 Correlations between discrete and serial

Individual differences across formats (within grade) pattern consistent with mean performance:

```
cor_serdis <- round(rbind(
  diag(cor( rsdw[,paste(c("logonsRT"),tasks,sep=".")], rsdw[,paste(c("logartic"),tasks,sep=".")], use="pa
  diag(cor( rsdw[,paste(c("logonsRT"),tasks,sep=".")], rsdw[,paste(c("logpautE"),tasks,sep=".")], use="pa
  diag(cor( rsdw[,paste(c("logonsRT"),tasks,sep=".")], rsdw[,paste(c("logarttE"),tasks,sep=".")], use="pa
  diag(cor( rsdw[,paste(c("logartic"),tasks,sep=".")], rsdw[,paste(c("logpautE"),tasks,sep=".")], use="pa
  diag(cor( rsdw[,paste(c("logartic"),tasks,sep=".")], rsdw[,paste(c("logarttE"),tasks,sep=".")], use="pa
  diag(cor( rsdw[,paste(c("logpautE"),tasks,sep=".")], rsdw[,paste(c("logarttE"),tasks,sep=".")], use="pa
),2)

cor_serdis <- by(rsdw,rsdw$grade,function(v){
  round(rbind(
    diag(cor( v[,paste(c("logonsRT"),tasks,sep=".")],
              v[,paste(c("logartic"),tasks,sep=".")],
              use="pairwise", method="spearman" )),
    diag(cor( v[,paste(c("logonsRT"),tasks,sep=".")],
              v[,paste(c("logpautE"),tasks,sep=".")],
              use="pairwise", method="spearman" )),
    diag(cor( v[,paste(c("logonsRT"),tasks,sep=".")],
              v[,paste(c("logarttE"),tasks,sep=".")],
              use="pairwise", method="spearman" )),
  )
})
```

```

diag(cor( v[,paste(c("logartic"),tasks,sep=".")],
           v[,paste(c("logpautE"),tasks,sep=".")],
           use="pairwise", method="spearman" )),
diag(cor( v[,paste(c("logartic"),tasks,sep=".")],
           v[,paste(c("logarttE"),tasks,sep=".")],
           use="pairwise", method="spearman" )),
diag(cor( v[,paste(c("logpautE"),tasks,sep=".")],
           v[,paste(c("logarttE"),tasks,sep=".")],
           use="pairwise", method="spearman" )),
),2)
})

for ( i in 1:3 ) {
colnames(cor_serdis[[i]]) <- tasks
rownames(cor_serdis[[i]]) <- c("Discrete onset - Discrete articulation",
                               "Discrete onset - Serial pause",
                               "Discrete onset - Serial articulation",
                               "Discrete articulation - Serial pause",
                               "Discrete articulation - Serial articulation",
                               "Serial pause - Serial articulation")
}
cor_serdis

## rsdw$grade: G1
##
## digit numword word dice object
## Discrete onset - Discrete articulation      0.31    0.31 0.45 0.30  0.10
## Discrete onset - Serial pause                0.48    0.53 0.84 0.60  0.33
## Discrete onset - Serial articulation         0.27    0.53 0.53 0.41  0.06
## Discrete articulation - Serial pause         0.00   -0.01 0.40 0.10 -0.05
## Discrete articulation - Serial articulation  0.32    0.39 0.77 0.39  0.41
## Serial pause - Serial articulation           0.18    0.34 0.51 0.14 -0.17
## -----
## rsdw$grade: G3
##
## digit numword word  dice object
## Discrete onset - Discrete articulation      0.10    0.36 0.41  0.29  0.13
## Discrete onset - Serial pause                0.41    0.49 0.73  0.62  0.50
## Discrete onset - Serial articulation         0.30    0.38 0.60  0.42  0.15
## Discrete articulation - Serial pause         0.02    0.19 0.22 -0.02  0.03
## Discrete articulation - Serial articulation  0.39    0.38 0.53  0.37  0.46
## Serial pause - Serial articulation           0.37    0.38 0.60  0.21 -0.02
## -----
## rsdw$grade: G5
##
## digit numword word  dice object
## Discrete onset - Discrete articulation      0.18    0.23 0.25 0.15  0.07
## Discrete onset - Serial pause                0.32    0.34 0.37 0.40  0.37
## Discrete onset - Serial articulation         0.33    0.24 0.34 0.26  0.24
## Discrete articulation - Serial pause         0.02    0.20 0.25 0.00  0.06
## Discrete articulation - Serial articulation  0.43    0.52 0.64 0.55  0.55
## Serial pause - Serial articulation           0.27    0.39 0.55 0.24 -0.13

```

In Grades 1 and 3, silent intervals are most highly correlated across formats (correlations between discrete onset latency and serial pause ranging between 0.48–0.84; excluding objects), followed by the correlation of discrete articulation with serial articulation time (0.32–0.77) and discrete onset latency with serial articulation (0.27–0.60). In Grade 5 correlations are generally a bit lower; spoken intervals are most highly

correlated (0.32–64), followed by silent intervals (0.32–0.40) and discrete onset latency with serial articulation (0.24–0.34).

5 Relationship between silent and spoken intervals in discrete tasks

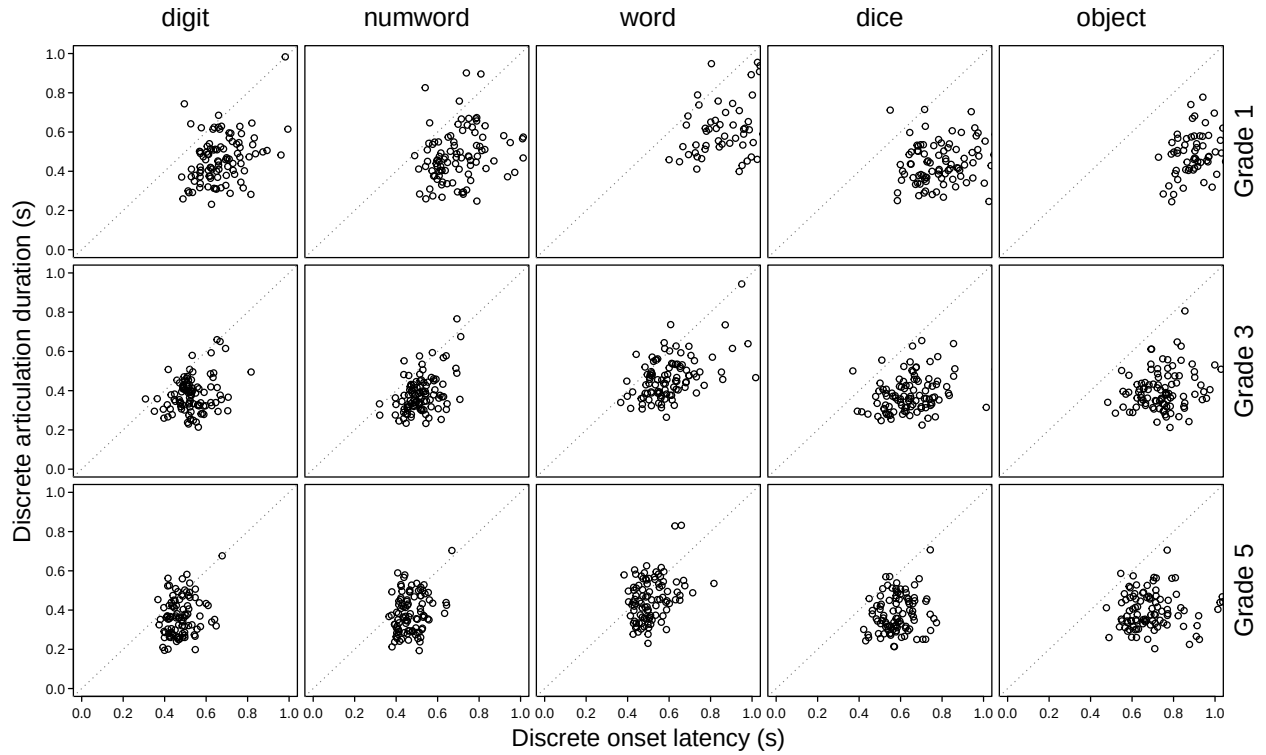
One way to visually examine how silent and spoken intervals compare to each other across children for each grade and task is to consider the location of points in a scatterplot against the diagonal.

```
par(mfrow=c(3,5))
par(oma=c(3.5,3.5,2,2))
par(mai=c(0.03,0.03,0.03,0.03))
par(mgp=c(2,0.5,0))
par(tcl= -0.3)

v1s <- "articsec"
v2s <- "onsRTsec"

for (grd in levels(rsdw$grade)) {
  for (tsk in tasks) {
    rsdw[,paste(v1s,tsk,sep=".")] <- rsdw[,paste("artic",tsk,sep=".")]/1000
    rsdw[,paste(v2s,tsk,sep=".")] <- exp(rsdw[,paste("logonsRT",tsk,sep=".")])/1000
    plot(as.formula(paste(paste(v1s,tsk,sep="."),paste(v2s,tsk,sep="."),sep=~")),
         rsdw,subset= grade==grd,
         xlim=c(0,1),ylim=c(0,1),xaxt="n",yaxt="n")
    if (grd=="G5") { axis(1) }
    if (tsk==tasks[1]) { axis(2,las=1) }
    abline(0,1,lty=3,col="grey50")
    if (grd=="G1") { mtext(tsk,3,line=1,cex=1.25)}
    if (tsk==tasks[length(tasks)]) { mtext(paste("Grade",substr(grd,2,2)),4,line=1,cex=1.25) }
  }
}

mtext("Discrete onset latency (s)", 1, outer=T, cex=1.2, line=2)
mtext("Discrete articulation duration (s)", 2, outer=T, cex=1.2, line=2.2)
```



The vast majority of points lie below the diagonal, consistent with a longer time needed for pre-articulation processing of the stimuli than for articulation. However, nonnegligible proportions of points lie around or above the diagonal for digits, number words, and words in Grade 5. These represent children who may have reached high levels of processing efficiency for these – most easily automatized – stimuli that they can process them faster than they can articulate them.

6 Association of serial advantage with discrete articulation

Using data from the discrete tasks only, processing time is measured by discrete onset latency and articulation time is measured by discrete articulation duration. We then compare these to the difference between discrete and serial total times (i.e., the serial advantage), calculated below.

```
for (tsk in tasks) {
  rsdw[paste("seradv",tsk,sep=".")] <- rsdw[paste("RT",tsk,sep=".")]/1000.0 -
    rsdw[paste("adjtim",tsk,sep=".")]/rsdw[paste("totcor",tsk,sep=".")]
}
seradv <- reshape( rsdw[,c("sID","grade", paste("seradv",tasks,sep="."))], direction="long",
  varying=paste("seradv",tasks,sep="."), sep=".", timevar="task", idvar=c("sID","grade"))
names(seradv)[4] <- "rawseradv"
seradv$task <- as.factor(seradv$task)
```

6.1 Graphical comparison

```
par(mfrow=c(1,3))
par(oma=c(2,2,0,0))
par(mai=c(0.2,0.2,0.2,0.1))
par(mgp=c(2,0.5,0))
par(tcl= -0.3)
par(cex.lab=1.5)
```

```

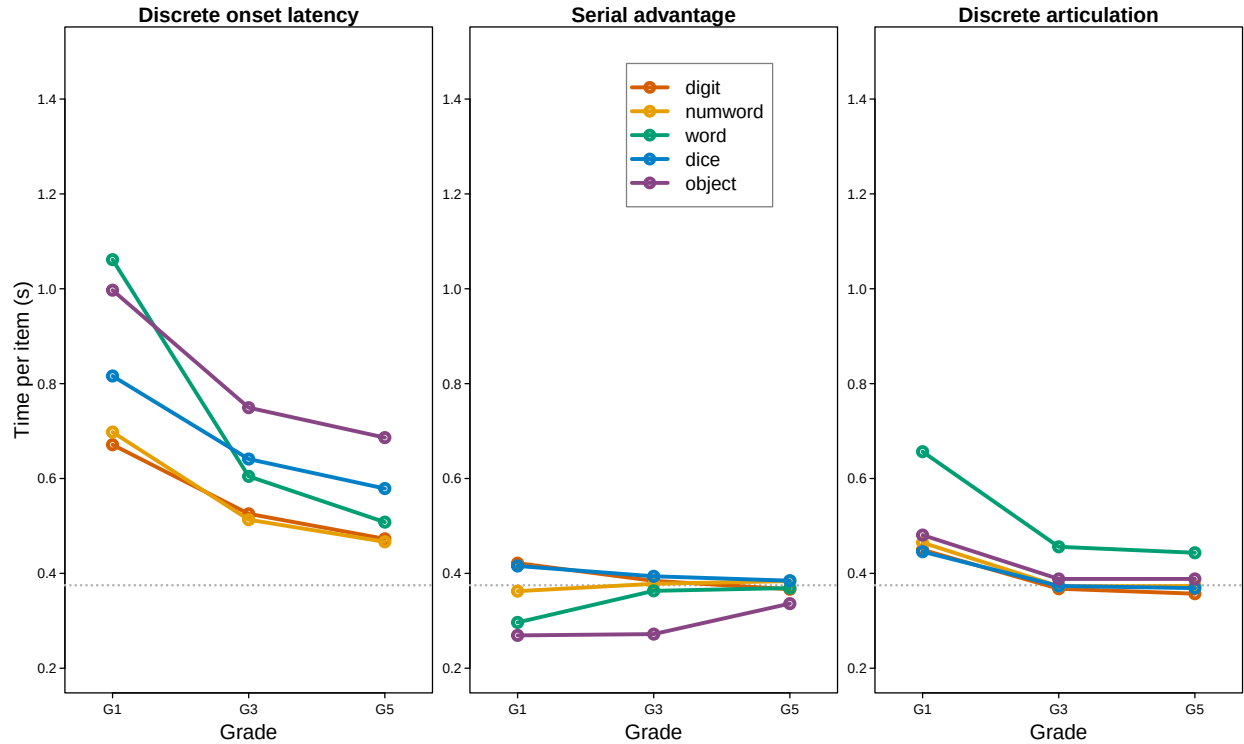
par(cex.main=1.5)

discons <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logonsRT",tasks,sep="."),collapse="),")",
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.2,1.5),xaxt="n",xlab="Grade",ylab="Time per item (s)",las=1,
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
abline(h=0.375,lty="11",lwd=2,col="grey70")
for (itsk in 1:length(tasks)) {
  tsk <- paste("logonsRT",tasks[itsk],sep=".")
  lines(1:3,discons[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,discons[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}

seradv <- aggregate( as.formula( paste("cbind(", paste(paste("seradv",tasks,sep="."),collapse="),")",
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.2,1.5),xaxt="n",xlab="Grade",ylab="",las=1,main="Serial ",
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
abline(h=0.375,lty="11",lwd=2,col="grey70")
for (itsk in 1:length(tasks)) {
  tsk <- paste("seradv",tasks[itsk],sep=".")
  lines(1:3,seradv[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,seradv[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}
legend(1.8,1.475,tasks,col=tcols,pch=21,lty=1,lwd=3,cex=1.45,box.col="grey50")

discart <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logartic",tasks,sep="."),collapse="),")",
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.2,1.5),xaxt="n",xlab="Grade",ylab="",las=1,main="Discrete ",
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
abline(h=0.375,lty="11",lwd=2,col="grey70")
for (itsk in 1:length(tasks)) {
  tsk <- paste("logartic",tasks[itsk],sep=".")
  lines(1:3,discart[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,discart[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}

```



It appears that the average serial advantage indeed approaches articulation time for presumably more efficient tasks (digits, number words, and dice) in Grades 3 and 5. According to the theoretical hypothesis leading to this prediction, this is consistent with efficient scheduling of temporally overlapped processing (cascading) in two stages that are mutually shielded from interference.

In contrast, discrete onset latencies clearly exceed the serial advantage, indicating that performance of these naming tasks is not – on average – articulation limited in these grades.

6.2 Statistical comparison of grade medians

```
options(width=110)
seradv_disart_tab <- numeric()
for (grd in levels(rsdw$grade)) {
  glines <- numeric()
  for (tsk in tasks) {
    seradvgt <- rsdw[rsdw$grade==grd,paste("seradv",tsk,sep=".")]
    disartgt <- exp(rsdw[rsdw$grade==grd,paste("logartic",tsk,sep=".")])/1000.0
    wilcox.test(seradvgt,disartgt,paired=T) -> wt
    line <- c(round(mean(seradvgt,na.rm=T),3), round(mean(disartgt,na.rm=T),3), wt$statistic, wt$p.value)
    glines <- rbind(glines,line)
  }
  colnames(glines) <- paste(c("seradv","disart","W","p"),grd,sep=".")
  seradv_disart_tab <- cbind(seradv_disart_tab,glines)
}
rownames(seradv_disart_tab) <- tasks

raw.pvals <- seradv_disart_tab[,substr(colnames(seradv_disart_tab),1,3)=="p.G"]
adj.pvals <- matrix(
  p.adjust(as.vector(raw.pvals),method="holm"),nrow=5,byrow=FALSE)
```

```
# unadjusted p values
seradv_disart_tab[,substr(colnames(seradv_disart_tab),1,3)=="p.G"] <- round(raw.pvals,3)
print(seradv_disart_tab)
```

	seradv.G1	disart.G1	W.G1	p.G1	seradv.G3	disart.G3	W.G3	p.G3	seradv.G5	disart.G5	W.G5	p.G5
## digit	0.421	0.464	1544	0.004	0.384	0.378	2755	0.430	0.365	0.369	2834	0.100
## numword	0.371	0.484	655	0.000	0.374	0.383	2476	0.737	0.384	0.385	2648	0.330
## word	0.302	0.682	107	0.000	0.362	0.467	506	0.000	0.366	0.455	653	0.000
## dice	0.421	0.462	1707	0.016	0.393	0.384	2646	0.552	0.380	0.380	2616	0.293
## object	0.270	0.496	436	0.000	0.273	0.402	429	0.000	0.338	0.398	953	0.000

```
# adjusted p values
seradv_disart_tab[,substr(colnames(seradv_disart_tab),1,3)=="p.G"] <- round(adj.pvals,3)
print(seradv_disart_tab)
```

	seradv.G1	disart.G1	W.G1	p.G1	seradv.G3	disart.G3	W.G3	p.G3	seradv.G5	disart.G5	W.G5	p.G5
## digit	0.421	0.464	1544	0.034	0.384	0.378	2755	1	0.365	0.369	2834	0.601
## numword	0.371	0.484	655	0.000	0.374	0.383	2476	1	0.384	0.385	2648	1.000
## word	0.302	0.682	107	0.000	0.362	0.467	506	0	0.366	0.455	653	0.000
## dice	0.421	0.462	1707	0.113	0.393	0.384	2646	1	0.380	0.380	2616	1.000
## object	0.270	0.496	436	0.000	0.273	0.402	429	0	0.338	0.398	953	0.000

The first table lists raw p values, whereas the second table lists p values that have been adjusted for multiple comparisons using the Holm method, which is less conservative than Bonferroni. A correction for multiple comparisons is not really appropriate here because it will bias the results toward non-significance. As we are not interested in detecting significant differences here, such a bias would work in favor of our hypothesis.

Consistent with the graph above, the difference between serial advantage and discrete articulation is not statistically significant for digits, dice, and number words in Grades 3 and 5. The difference is substantial and statistically significant for objects and words in every grade.

6.3 Serial advantage vs. articulation: individual differences (all children)

The previous illustration concerned grade averages. What about individual differences? The predictions are meant to apply to individual children, even though measurement noise can be expected to obscure the relationship to some extent.

On each scatterplot, the corresponding Spearman's ρ is displayed in blue, and the unstandardized β regression coefficient is displayed in violet/purple.

```
par(mfrow=c(3,5))
par(oma=c(3.5,3.5,2,2))
par(mai=c(0.03,0.03,0.03,0.03))
par(mgp=c(2,0.5,0))
par(tcl= -0.3)

v1s <- "articsec"
v2s <- "seradv"

for (grd in levels(seradv$grade)) {
  for (tsk in tasks) {
    rsdw[,paste(v1s,tsk,sep=".")] <- rsdw[,paste("artic",tsk,sep=".")]/1000
    plot(as.formula(paste(paste(v1s,tsk,sep="."),paste(v2s,tsk,sep="."),sep="~")),
         rsdw,subset= grade==grd,
         xlim=c(0,1),ylim=c(0,1),xaxt="n",yaxt="n")
    if (grd=="G5") { axis(1) }
    if (tsk==tasks[1]) { axis(2,las=1) }
  }
}
```

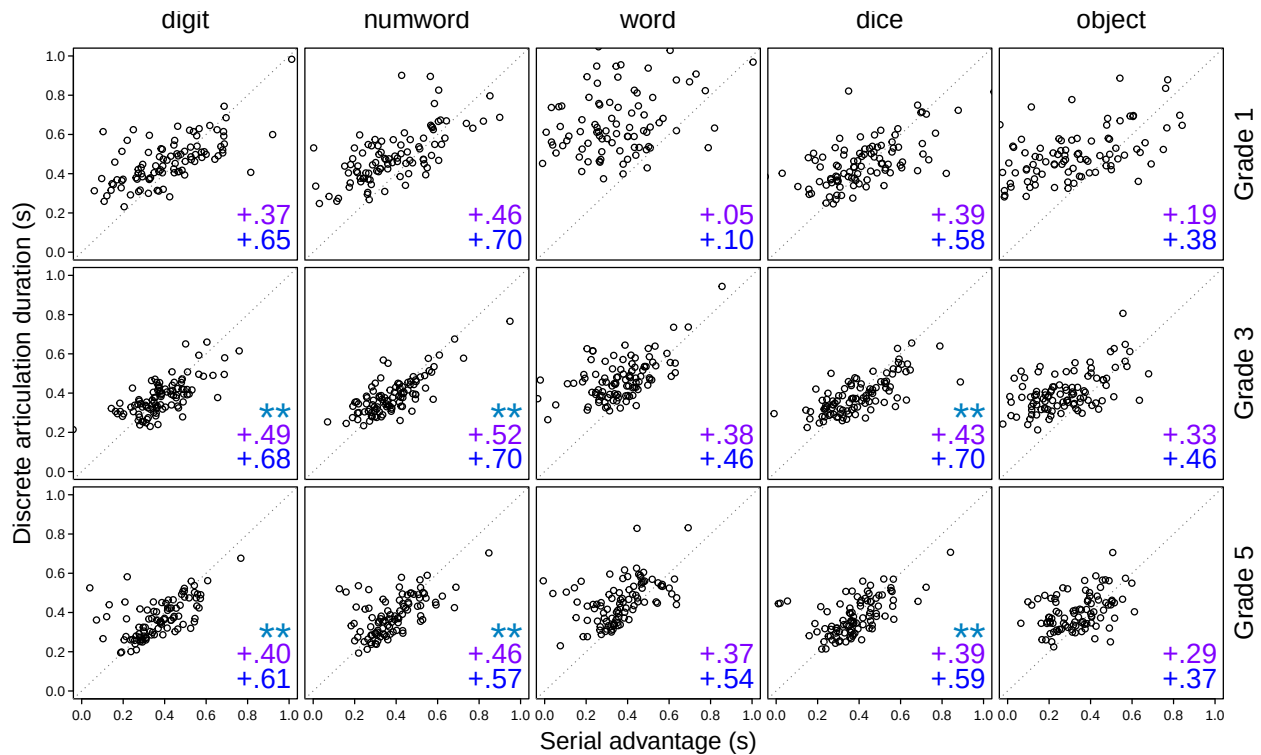
```

abline(0,1,lty=3,col="grey50")

cstr <- signs(
  cor(rsdw[rsdw$grade==grd,paste(v1s,tsk,sep=".")],
      rsdw[rsdw$grade==grd,paste(v2s,tsk,sep=".")],
      method="spearman",use="pairwise"),
  accuracy = .01, trim_leading_zeros = TRUE, add_plusses=TRUE)
text(1.05,0.05,cstr,cex=2,pos=2,col="blue")
lmbetastr <- signs(
  coef(lm(
    as.formula(paste(paste(v1s,tsk,sep="."),paste(v2s,tsk,sep="."),sep=~")),
    data=rsdw,subset= grade==grd))[2],
  accuracy = .01, trim_leading_zeros = TRUE, add_plusses=TRUE)
text(1.05,0.175,lmbetastr,cex=2,pos=2,col="#8000FF")
v1d <- rsdw[rsdw$grade==grd,paste(v1s,tsk,sep=".")]
v2d <- rsdw[rsdw$grade==grd,paste(v2s,tsk,sep=".")]
wilcox.test(v1d,v2d,paired=TRUE) -> wt
if (wt$p.value >= 0.05) { text(1.05,0.25,"**",cex=3,pos=2,col="#0080C0") }

if (grd=="G1") { mtext(tsk,3,line=1,cex=1.25)}
if (tsk==tasks[length(tasks)]) { mtext(paste("Grade",substr(grd,2,2)),4,line=1,cex=1.25) }
}
}
mtext("Serial advantage (s)", 1, outer=T, cex=1.2, line=2)
mtext("Discrete articulation duration (s)", 2, outer=T, cex=1.2, line=2.2)

```



Indeed, the observed correlations are sizeable across grades and tasks; and, consistent with the results of comparing means, correlations are clearly smallest for the least efficiently processed stimuli, namely words before Grade 5 and objects across grades.

7 Association of total serial time with discrete onset latency

If participants in serial tasks begin to process the next item as soon as they are ready to utter the current item, and if processing takes longer than articulation (so that there are silent intervals), then the sum of articulation (of the current item) and pause (silence before the next item) should equal the processing time for the item (i.e., discrete onset latency).

7.1 Graph of serial total time vs. discrete onset latency

```
par(mfrow=c(1,3))
par(oma=c(2,2,0,0))
par(mai=c(0.3,0.4,0.4,0.1))
par(mgp=c(2.25,0.5,0))
par(tcl= -0.3)
par(cex.lab=1.5)
par(cex.main=1.5)

ymx <- 1.5
lny <- 0.628

discons <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logonsRT",tasks,sep="."),collapse=","),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.2,ymx),xaxt="n",xlab="",ylab="Time per item (s)",las=1,m
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
abline(h=lny,lty="11",lwd=2,col="grey70")
for (itsk in 1:length(tasks)) {
  tsk <- paste("logonsRT",tasks[itsk],sep=".")
  lines(1:3,discons[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,discons[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}

serartpau <- aggregate(
  as.formula(
    paste("cbind(",
      paste(paste(
        paste("itmarttE",tasks,sep="."),
        paste("itmpautE",tasks,sep="."),sep="+"
      ),collapse=","),
      "~ grade", sep="") ),
  rsdw, mean, na.rm=T )[, -1] # remove grade column
colnames(serartpau) <- paste("serartpau",tasks,sep=".")
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.2,ymx),xaxt="n",xlab="Grade",ylab="",las=1,main="Serial p
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
abline(h=lny,lty="11",lwd=2,col="grey70")
for (itsk in 1:length(tasks)) {
  tsk <- paste("serartpau",tasks[itsk],sep=".")
  lines(1:3,serartpau[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,serartpau[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}

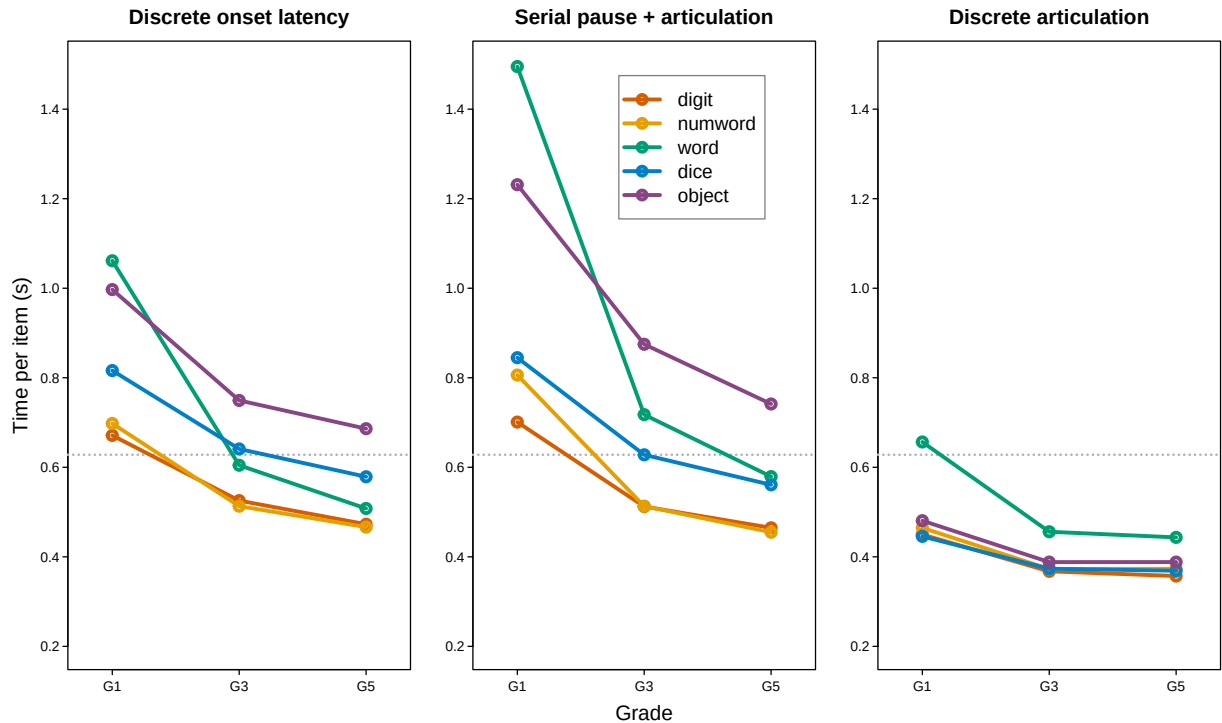
legend(1.8,ymx-0.025,tasks,col=tcols,pch=21,lty=1,lwd=3,cex=1.45,box.col="grey50")

discart <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logartic",tasks,sep="."),collapse=","),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.2,ymx),xaxt="n",xlab="",ylab="",las=1,main="Discrete art.
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
```

```

abline(h=lny,lty="11",lwd=2,col="grey70")
for (itsk in 1:length(tasks)) {
  tsk <- paste("logartic",tasks[itsk],sep=".")
  lines(1:3,discart[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,discart[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}

```



There is remarkable consistency in these graphs even though they are measured entirely independently. Serial articulation+pause seems to closely follow discrete onset latency, at least for the less demanding tasks. The relationship breaks down for words and objects, especially in earlier grades, suggesting lack of efficient scheduling in those cases.

7.2 Correlations between serial pause+articulation and discrete onset latency

If the one-overlap hypothesis is on the right track it should hold at the individual level, not just for means. The following graphs plot this relationship for each task and grade.

```

for (tsk in tasks) {
  rsdw[,paste("serartpau",tsk,sep=".")] <- rsdw[,paste("itmarttE",tsk,sep=".")] + rsdw[,paste("itmpautE",tsk,sep=".")]
  rsdw[,paste("onsRTsec",tsk,sep=".")] <- rsdw[,paste("onsRT",tsk,sep=".")] / 1000.0
  # sum of articulation and pause only for those who have mean pause >= .05
  rsdw[,paste("serartpau2",tsk,sep=".")] <- ifelse ( rsdw[,paste("itmpautE",tsk,sep=".")] < 0.05, NA, rsdw[,paste("serartpau",tsk,sep=".")] + rsdw[,paste("itmpautE",tsk,sep=".")] )
}

par(mfrow=c(3,5))
par(oma=c(3.5,3.75,2,1.75))
par(mai=c(0.03,0.03,0.03,0.03))
par(mgp=c(2,0.5,0))
par(tcl= -0.3)

```

```

v1s <- "onsRTsec"
v2s <- "serartpau"

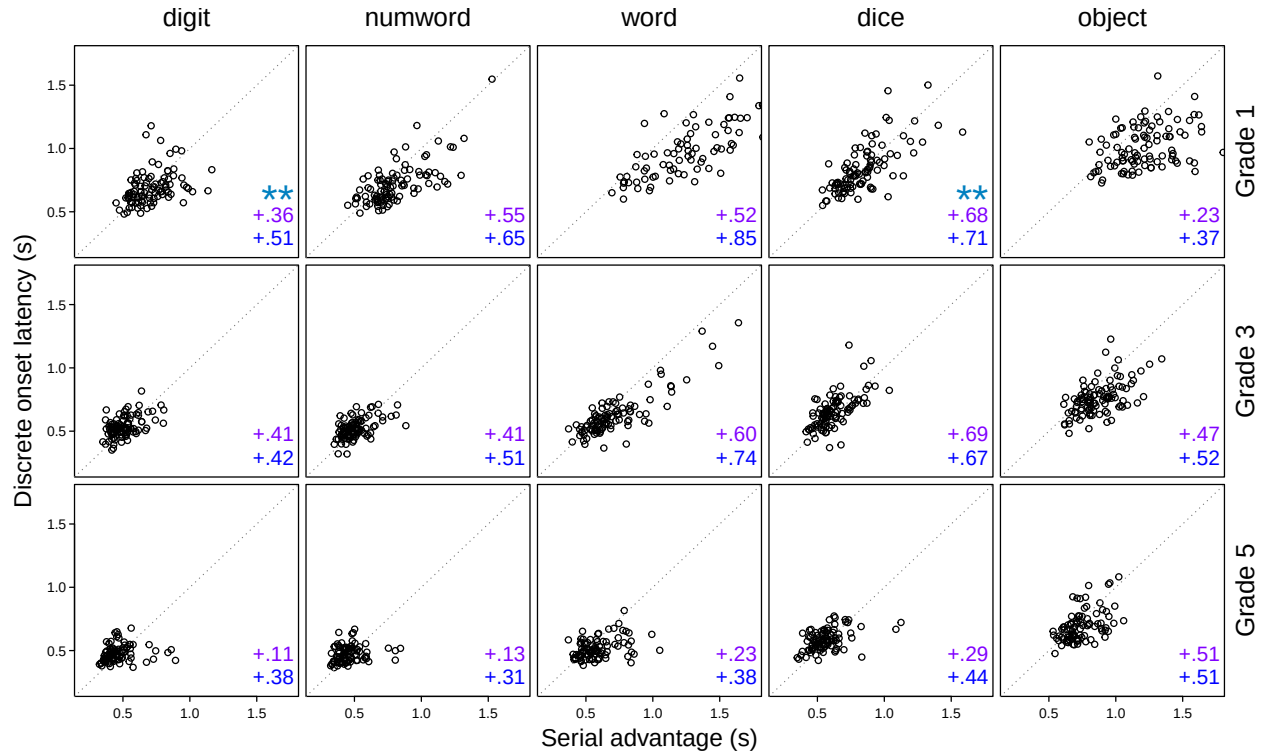
for (grd in levels(rsdw$grade)) {
  for (tsk in tasks) {
    plot(as.formula(paste(paste(v1s,tsk,sep=". "),paste(v2s,tsk,sep=". "),sep="~")),
         rsdw,subset= grade==grd,
         xlim=c(0.20,1.75),ylim=c(0.20,1.75),
         xaxt="n",yaxt="n")
    if (grd=="G5") { axis(1) }
    if (tsk==tasks[1]) { axis(2,las=1) }
    abline(0,1,lty=3,col="grey50")

    cstr <- signs(
      cor(rsdw[rsdw$grade==grd,paste(v1s,tsk,sep=". ")],
          rsdw[rsdw$grade==grd,paste(v2s,tsk,sep=". ")],
          method="spearman",use="pairwise"),
      accuracy = .01, trim_leading_zeros = TRUE, add_plusses=TRUE)
    text(1.84,0.275,cstr,cex=1.5,pos=2,col="blue")
    lmbetastr <- signs(
      coef(lm(
        as.formula(paste(paste(v1s,tsk,sep=". "),paste(v2s,tsk,sep=". "),sep="~")),
        data=rsdw,subset= grade==grd))[2],
      accuracy = .01, trim_leading_zeros = TRUE, add_plusses=TRUE)
    text(1.84,0.460,lmbetastr,cex=1.5,pos=2,col="#8000FF")
    v1d <- rsdw[rsdw$grade==grd,paste(v1s,tsk,sep=". ")]
    v2d <- rsdw[rsdw$grade==grd,paste(v2s,tsk,sep=". ")]
    wilcox.test(v1d,v2d,paired=TRUE) -> wt
    if (wt$p.value >= 0.05) { text(1.84,0.560,"**",cex=3,pos=2,col="#0080C0") }

    if (grd=="G1") { mtext(tsk,3,line=1,cex=1.25)}
    if (tsk==tasks[length(tasks)]) { mtext(paste("Grade",substr(grd,2,2)),4,line=1,cex=1.25) }
  }
}

mtext("Serial advantage (s)", 1, outer=T, cex=1.2, line=2)
mtext("Discrete onset latency (s)", 2, outer=T, cex=1.2, line=2.2)

```



These correlations are substantial, consistent with the hypothesis. Interestingly, they are weakest for the most efficiently processed cases, indicating that the one-overlap assumption eventually breaks down as skill increases.

7.3 Statistical comparison between serial pause+articulation and discrete onset latency

These means can be statistically compared using a (nonparametric) Wilcoxon signed ranks test; W and p are listed below beside the corresponding pair.

```
options(width=110)
serap_disons_tab <- numeric()
for (grd in levels(rsdw$grade)) {
  glines <- numeric()
  for (tsk in tasks) {
    #print(paste(grd,tsk))
    serapgt <- rsdw[rsdw$grade==grd,paste("itmarttE",tsk,sep=".")] + rsdw[rsdw$grade==grd,paste("itmpa",tsk,sep=".")]
    disonsgt <- rsdw[rsdw$grade==grd,paste("onsRT",tsk,sep=".")] / 1000.0
    wilcox.test(serapgt,disonsgt,paired=T) -> wt
    line <- c(round(mean(serapgt,na.rm=T),3), round(mean(disonsgt,na.rm=T),3), wt$statistic, wt$p.value)
    glines <- rbind(glines,line)
  }
  colnames(glines) <- paste(c("serapgt","disonsgt","W","p"),grd,sep=".")
  serap_disons_tab <- cbind(serap_disons_tab,glines)
}
rownames(serap_disons_tab) <- tasks

raw.pvals <- serap_disons_tab[,substr(colnames(serap_disons_tab),1,3)=="p.G"]
adj.pvals <- matrix(
  p.adjust(as.vector(raw.pvals),method="holm"),nrow=5,byrow=FALSE)
```

```
# unadjusted p values
serap_disons_tab[,substr(colnames(serap_disons_tab),1,3)=="p.G"] <- round(raw.pvals,3)
print(serap_disons_tab)
```

```
##          serapgt.G1 disonsgt.G1 W.G1  p.G1 serapgt.G3 disonsgt.G3 W.G3  p.G3 serapgt.G5 disonsgt.G5 W
## digit          0.708          0.684 2743 0.130          0.509          0.531 1707 0.005          0.464          0.476 1
## numword         0.805          0.722 3808 0.000          0.507          0.519 1995 0.049          0.455          0.470 1
## word           1.514          1.110 4570 0.000          0.712          0.622 4338 0.000          0.580          0.513 3
## dice           0.842          0.845 2421 0.874          0.623          0.653 1578 0.002          0.564          0.584 1
## object         1.233          1.013 4415 0.000          0.871          0.759 4469 0.000          0.739          0.695 3
```

```
# adjusted p values
serap_disons_tab[,substr(colnames(serap_disons_tab),1,3)=="p.G"] <- round(adj.pvals,3)
print(serap_disons_tab)
```

```
##          serapgt.G1 disonsgt.G1 W.G1  p.G1 serapgt.G3 disonsgt.G3 W.G3  p.G3 serapgt.G5 disonsgt.G5 W
## digit          0.708          0.684 2743 0.260          0.509          0.531 1707 0.025          0.464          0.476 1
## numword         0.805          0.722 3808 0.000          0.507          0.519 1995 0.148          0.455          0.470 1
## word           1.514          1.110 4570 0.000          0.712          0.622 4338 0.000          0.580          0.513 3
## dice           0.842          0.845 2421 0.874          0.623          0.653 1578 0.012          0.564          0.584 1
## object         1.233          1.013 4415 0.000          0.871          0.759 4469 0.000          0.739          0.695 3
```

The difference is typically quite small, 10–50 msec, for Grades 3 and 5, with the exception of words. It is not statistically significant only for digits and dice in Grade 1.

Here is the table of differences (rounded to the nearest millisecond), to facilitate comparisons:

```
serap_disons_diffTAB <- numeric()
for (grd in levels(rsdw$grade)) {
  glines <- numeric()
  for (tsk in tasks) {
    serapgt <- (rsdw[rsdw$grade==grd,paste("itmarttE",tsk,sep=".")] + rsdw[rsdw$grade==grd,paste("itmp",tsk,sep=".")])
    disonsgt <- rsdw[rsdw$grade==grd,paste("onsRT",tsk,sep=".")]
    line <- c( round( mean(serapgt,na.rm=T) - mean(disonsgt,na.rm=T), 0 ) )
    glines <- rbind(glines,line)
  }
  colnames(glines) <- paste("diff",grd,sep=".")
  serap_disons_diffTAB <- cbind(serap_disons_diffTAB,glines)
}
rownames(serap_disons_diffTAB) <- tasks
print(serap_disons_diffTAB)
```

```
##          diff.G1 diff.G3 diff.G5
## digit          24      -23      -12
## numword         83      -11      -15
## word          404         89         67
## dice           -3       -30       -20
## object         220        112         44
```

Interestingly, the sum of serial spoken and articulation interval duration consistently exceeds discrete onset latency for words and objects (if greatly diminishing with grade), indicating that – in these most demanding tasks – children take longer in the serial version than predicted from discrete onset latency under the maximal efficiency hypothesis. In contrast, the difference is negative and significant for digits, dice, and number words in Grades 3 and (especially) 5, indicating that efficient scheduling can go beyond the one-overlap assumption in these cases.

8 Associations of naming response components with reading measures

The following table displays descriptives for the reading measures (untransformed scores), for illustration.

8.1 Descriptives for reading measures

```
rawreadmeas <- c("wrldacc","pwrldacc","wflut","pwflu")
rmnice <- c("Word accuracy","Pseud. accuracy","Word fluency","Pseud. fluency")

descread <- cbind(
  as.vector(t(aggregate(
    as.formula( paste("cbind(", paste(rawreadmeas, collapse=","), ") ~ grade", sep="") ),
    rsdw, drop=FALSE, na.action=na.pass, FUN=mean, na.rm=T )[, -1])),
  as.vector(t(aggregate(
    as.formula( paste("cbind(", paste(rawreadmeas, collapse=","), ") ~ grade", sep="") ),
    rsdw, drop=FALSE, na.action=na.pass, FUN=sd, na.rm=T )[, -1])),
  as.vector(t(aggregate(
    as.formula( paste("cbind(", paste(rawreadmeas, collapse=","), ") ~ grade", sep="") ),
    rsdw, drop=FALSE, na.action=na.pass, FUN=min, na.rm=T )[, -1])),
  as.vector(t(aggregate(
    as.formula( paste("cbind(", paste(rawreadmeas, collapse=","), ") ~ grade", sep="") ),
    rsdw, drop=FALSE, na.action=na.pass, FUN=max, na.rm=T )[, -1]))
)
descread <- round( cbind( descread[1:4,], descread[5:8,], descread[9:12,] ), 1 )
colnames(descread) <- paste( rep(paste("G",c(1,3,5),sep=""),each=4),
                             rep(c("M","SD","min","max"),3),
                             sep="." )
rownames(descread) <- rmnice
descread
```

##		G1.M	G1.SD	G1.min	G1.max	G3.M	G3.SD	G3.min	G3.max	G5.M	G5.SD	G5.min	G5.max
##	Word accuracy	69.4	7.1	46	80	75.2	4.9	55	80	78.2	2.7	67	80
##	Pseud. accuracy	23.6	4.7	15	37	28.5	7.4	14	38	33.2	5.5	17	38
##	Word fluency	30.7	10.3	11	62	55.8	11.6	27	84	71.6	11.4	42	95
##	Pseud. fluency	18.1	5.2	6	34	28.7	6.7	14	49	36.6	7.3	20	54

This is the content of Table 6 in the manuscript.

8.2 Tables of correlations

To connect with previous research on “RAN components”, here are the nonparametric correlations (Spearman’s ρ) between reading measures and silent and spoken intervals in discrete and serial naming tasks.

```
rmeas <- c("logwacc","sqpwacc","wflut","pwflu")
nmeas <- c("logonsRT","logartic","logpautE","logarttE")
by(rsdw,rsdw$grade,function(x){round(cor(x[,c(rmeas,
                                             apply(expand.grid(nmeas,tasks),1,function(x){paste(x[1],x
                                             ]),
                                             method="spearman",use="pairwise") [1:4,-c(1:4)]
                                             ,2)} ) -> rncor
for (grd in levels(rsdw$grade)) {
  cat(paste("\n***",grd,"\n"))
  print(t(rncor[[grd]]))
}
```

```

##
## *** G1
##          logwacc sqpwacc wflut pwflu
## logonsRT.digit      0.27  -0.10  0.35  0.47
## logartic.digit      0.10  -0.07  0.15  0.18
## logpautE.digit      0.21  -0.02  0.35  0.46
## logarttE.digit      0.02   0.07  0.36  0.47
## logonsRT.numword    0.46  -0.02  0.58  0.54
## logartic.numword    0.18   0.04  0.25  0.30
## logpautE.numword    0.32  -0.01  0.64  0.46
## logarttE.numword    0.24   0.05  0.57  0.46
## logonsRT.word       0.33   0.01  0.74  0.68
## logartic.word       0.20  -0.03  0.51  0.44
## logpautE.word       0.32  -0.06  0.78  0.70
## logarttE.word       0.12   0.02  0.62  0.53
## logonsRT.dice       0.28   0.04  0.44  0.46
## logartic.dice       0.12  -0.02  0.21  0.26
## logpautE.dice       0.27  -0.02  0.35  0.44
## logarttE.dice      -0.07   0.08  0.26  0.29
## logonsRT.object     0.23   0.18  0.27  0.27
## logartic.object     0.10   0.03  0.24  0.26
## logpautE.object     0.16   0.15  0.32  0.21
## logarttE.object    -0.12   0.14  0.14  0.15
##
## *** G3
##          logwacc sqpwacc wflut pwflu
## logonsRT.digit      0.26   0.43  0.45  0.47
## logartic.digit     -0.02   0.07  0.26  0.12
## logpautE.digit      0.19   0.23  0.45  0.47
## logarttE.digit      0.14   0.04  0.52  0.49
## logonsRT.numword    0.28   0.32  0.50  0.49
## logartic.numword    0.00   0.05  0.30  0.18
## logpautE.numword    0.32   0.33  0.59  0.56
## logarttE.numword    0.25   0.22  0.65  0.60
## logonsRT.word       0.38   0.44  0.74  0.67
## logartic.word       0.05   0.03  0.39  0.26
## logpautE.word       0.44   0.37  0.77  0.66
## logarttE.word       0.27   0.21  0.72  0.59
## logonsRT.dice       0.28   0.34  0.46  0.51
## logartic.dice      -0.05   0.00  0.23  0.13
## logpautE.dice       0.22   0.26  0.47  0.48
## logarttE.dice       0.16   0.19  0.45  0.44
## logonsRT.object     0.11   0.24  0.27  0.30
## logartic.object    -0.04   0.01  0.20  0.06
## logpautE.object     0.14   0.15  0.33  0.33
## logarttE.object     0.19   0.16  0.32  0.25
##
## *** G5
##          logwacc sqpwacc wflut pwflu
## logonsRT.digit      0.10   0.02  0.31  0.24
## logartic.digit      0.02   0.16  0.12  0.06
## logpautE.digit      0.19   0.11  0.43  0.43
## logarttE.digit      0.09   0.12  0.43  0.40
## logonsRT.numword    0.11   0.06  0.38  0.32

```

## logartic.numword	0.07	0.20	0.20	0.13
## logpautE.numword	0.16	0.12	0.60	0.51
## logarttE.numword	0.13	0.16	0.50	0.44
## logonsRT.word	0.17	0.05	0.54	0.44
## logartic.word	0.07	0.15	0.27	0.16
## logpautE.word	0.22	0.20	0.57	0.55
## logarttE.word	0.15	0.18	0.53	0.46
## logonsRT.dice	0.10	0.07	0.28	0.24
## logartic.dice	0.04	0.19	0.16	0.12
## logpautE.dice	0.23	0.09	0.31	0.36
## logarttE.dice	0.07	0.16	0.34	0.24
## logonsRT.object	0.16	0.05	0.34	0.33
## logartic.object	0.04	0.19	0.21	0.13
## logpautE.object	0.23	0.11	0.25	0.29
## logarttE.object	-0.01	0.06	0.14	0.11

8.3 Graphical display of correlations

To facilitate comparisons, the above correlations are displayed graphically below. In the following graph, each row of panels concerns one reading measure (word reading accuracy, pseudoword reading accuracy, word list reading fluency, and pseudoword list reading fluency). Each column of panels concerns one type of naming stimulus (digits, number words, words, dice, and objects). Within each panel, there are four bar triplets: The two triplets on the left (lighter colors) concern discrete naming tasks and the two triplets on the right (darker colors) concern serial naming tasks. For each pair of triplets, silent intervals are on the left and spoken intervals on the right. Each triplet displays correlation coefficients between the corresponding reading and naming measures for three groups, namely Grades 1, 3, and 5, distinguished by increasing shading density.

```
#pdf("corread.pdf",width=10,height=7)
par(mfcol=c(4,5))
par(oma=c(7,4,1,2))
par(mai=c(0.05,0.05,0.05,0.05))
par(mgp=c(2,0.6,0))
par(tcl= -0.2)

# Run a dummy plot to obtain the horizontal coordinates for placement
barplot(rbind(rncor[["G1"]][1,paste(nmeas,c(tasks[1]),sep=".")],
              rncor[["G1"]][1,paste(nmeas,c(tasks[1]),sep=".")],
              rncor[["G1"]][1,paste(nmeas,c(tasks[1]),sep=".")]),
        ylim=c(0,1),width=0.2,beside=TRUE,plot=FALSE) -> mp

for (it in 1:length(tasks)) {

  tsk <- tasks[it]

  for (ir in 1:length(rmeas)) {

    rms <- rmeas[ir]

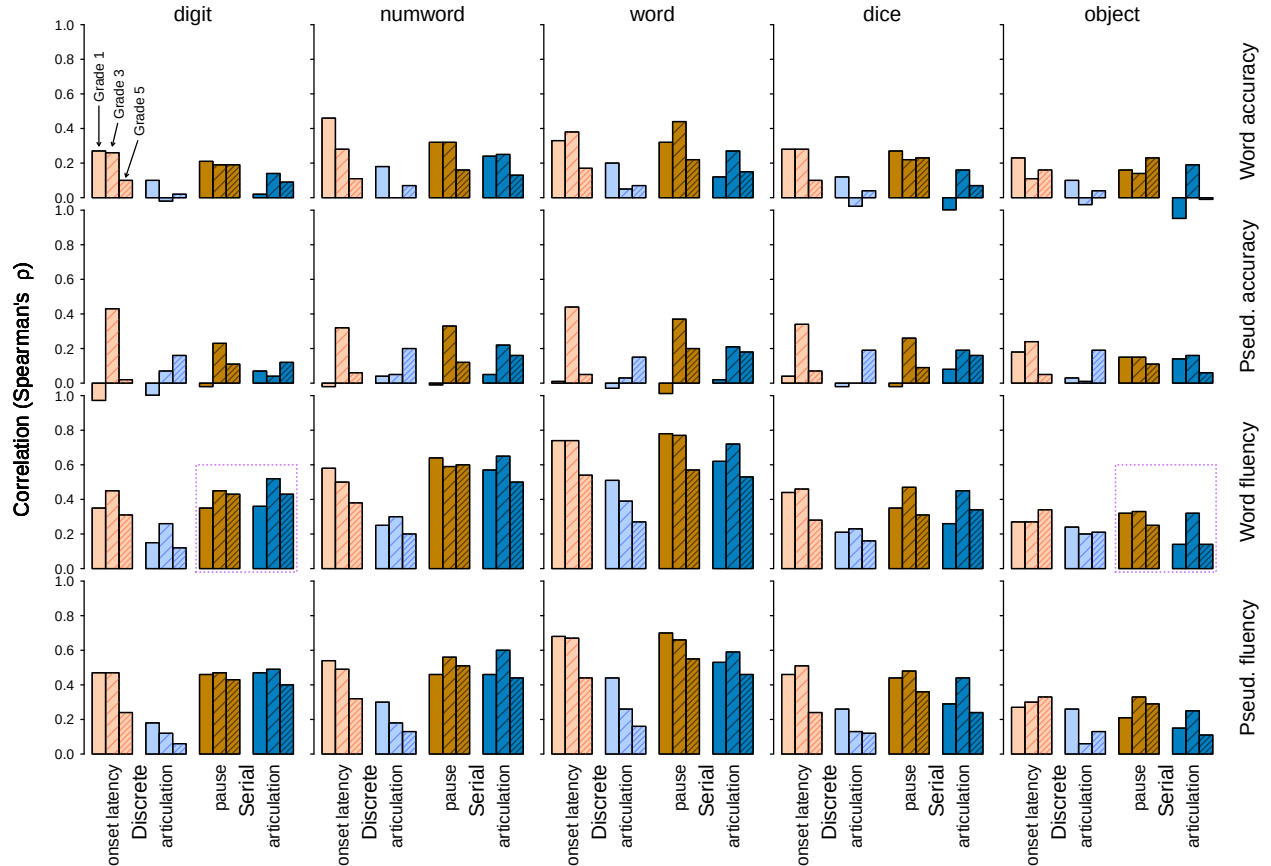
    # First plot the bars
    barplot(
      rbind(rncor[["G1"]][rms,paste(nmeas,c(tsk),sep=".")],
            rncor[["G3"]][rms,paste(nmeas,c(tsk),sep=".")],
            rncor[["G5"]][rms,paste(nmeas,c(tsk),sep=".")]),
```

```

ylim=c(0,1),axes=FALSE,axisnames=FALSE,xpd=NA,
col=rep(c("#FFD0B0","#B0D0FF","#C08000","#0080C0"),each=3),
width=0.2,beside=TRUE)
# Then draw the shading lines on top
barplot(
  rbind(rncor[["G1"]][rms,paste(nmeas,c(tsk),sep=".")],
        rncor[["G3"]][rms,paste(nmeas,c(tsk),sep=".")],
        rncor[["G5"]][rms,paste(nmeas,c(tsk),sep=".")]),
  ylim=c(0,1),axes=FALSE,axisnames=FALSE,xpd=NA,
  col=rep(c("#FF9070","#7090FF","#604000","#004060"),each=3),density=rep(c(0,15,30),4),
  width=0.2,beside=TRUE,add=TRUE)

# Axes, legends, etc.
if (ir==4) {
  axis(1,at=c(mean(mp[2,c(1,2)]),mean(mp[2,c(3,4)])),labels=c("Discrete","Serial"),las=2,tick=FALSE)
  axis(1,at=mp[2,],labels=c("onset latency","articulation","pause","articulation"),
      las=2,tick=FALSE,cex.axis=1.2,xpd=NA)
} else {
  axis(1,labels=NA,tick=FALSE)
}
if (ir==1) {
  mtext(tsk,3)
  if (it==1) {
    text(mp[1,1],0.7,"Grade 1",adj=c(0.5,0.5),srt=90,cex=1)
    text(mp[2,1]+0.1,0.6,"Grade 3",adj=c(0.5,0.5),srt=90,cex=1)
    text(mp[3,1]+0.2,0.5,"Grade 5",adj=c(0.5,0.5),srt=90,cex=1)
    arrows(mp[1,1],0.53,mp[1,1],0.30,length=0.03)
    arrows(mp[2,1]+0.08,0.43,mp[2,1],0.30,length=0.03)
    arrows(mp[3,1]+0.18,0.33,mp[3,1],0.12,length=0.03)
  }
}
if (ir==3) {
  if (it %in% c(1,5)) { rect(mp[1,3]-0.15,-0.02,mp[3,4]+0.15,0.6,col=NA,border="#D080FF",lwd=1.5,lt)
  }
  if (it==5) { mtext(rmnice[ir],4,line=1.25) }
  if (it==1) {
    axis(2,las=1)
  } else {
    axis(2,labels=NA)
  }
}
mtext(expression(paste("Correlation (Spearman's ",rho,")"),sep=""),2,outer=TRUE,line=2.5)
}

```



#dev.off()

In general, and as expected from the literature, we see that (a) reading fluency measures are predicted much more strongly than reading accuracy, and (b) a bit more strongly from digit naming (leftmost column) than from object naming (rightmost column). (c) The duration of spoken intervals from discrete naming (light blue bars) is the weakest predictor of reading measures across grades and tasks; and (d) the duration of silent intervals from discrete naming (onset latency; pink bars) is almost as good a predictor of reading measures as the duration of silent intervals from serial tasks (pause time; dark goldenrod bars). (e) Naming tasks involving print (that is, words and number words) are, naturally, the stronger predictors of reading measures.

The correlations most nearly (though not exactly) comparable with the literature are enclosed in lilac dashed boxes on the third row: they concern correlations between word list reading fluency and the “RAN components” of digit and object naming. We can see that silent and spoken intervals from digit naming (leftmost column) are associated with fluency to about the same extent, around 0.4. Associations with intervals from object naming (rightmost column) are lower, and less strong for spoken than for silent intervals.

9 Session information

```
date()
```

```
## [1] "Fri Mar 8 13:46:17 2024"
```

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16 ucrt)
```

```
## Platform: x86_64-w64-mingw32/x64 (64-bit)
```

```

## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=Norwegian Bokmål_Norway.utf8 LC_CTYPE=Norwegian Bokmål_Norway.utf8
## [3] LC_MONETARY=Norwegian Bokmål_Norway.utf8 LC_NUMERIC=C
## [5] LC_TIME=Norwegian Bokmål_Norway.utf8
##
## time zone: Europe/Berlin
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] signs_0.1.2      emmeans_1.8.6    multcomp_1.4-24 TH.data_1.1-2    MASS_7.3-60      survival_3.5-5
## [7] mvtnorm_1.2-2    nlme_3.1-162     psych_2.3.3
##
## loaded via a namespace (and not attached):
## [1] cli_3.6.1          knitr_1.43         rlang_1.1.1        xfun_0.39          estimability_1.4.1
## [6] highr_0.10         xtable_1.8-4       zoo_1.8-12         colorspace_2.1-0   htmltools_0.5.5
## [11] scales_1.2.1       rmarkdown_2.22     grid_4.3.1         munsell_0.5.0      evaluate_0.21
## [16] fastmap_1.1.1      lifecycle_1.0.3    yaml_2.3.7         compiler_4.3.1     codetools_0.2-19
## [21] coda_0.19-4        sandwich_3.0-2     rstudioapi_0.14    lattice_0.21-8     digest_0.6.31
## [26] R6_2.5.1           mnormt_2.1.1       parallel_4.3.1     splines_4.3.1      Matrix_1.5-4.1
## [31] tools_4.3.1

```