

Silent and spoken intervals in naming: Serial advantage

Athanassios Protopapas

7 February 2023

Contents

1	Introduction	2
2	Read in data	2
3	Differences between discrete and serial naming rates	2
3.1	Differences in total task time	2
3.1.1	Graphs of total times per item	2
3.1.2	Graphs of rates (inverse-transformed times)	4
4	The serial advantage	6
4.1	Calculation and distributions	6
4.2	Serial advantage predictions	7
4.3	Serial advantage vs. discrete silent and spoken intervals	7
4.3.1	Graphical comparison	7
4.3.2	Statistical comparison	9
4.4	Serial advantage vs. serial silent and spoken intervals	10
4.4.1	Graphical comparison	10
4.4.2	Statistical comparison	12
4.5	Serial advantage vs. articulation: individual differences (all children)	12
4.5.1	Correlations between serial advantage and discrete articulation duration	13
4.5.2	Correlations between serial advantage and discrete onset latency	14
4.5.3	Correlations between serial advantage and serial articulation times	15
4.6	Serial advantage vs. articulation: individual differences (exclude children with negligible pause times)	17
4.6.1	Correlations between serial advantage and discrete articulation duration	17
4.6.2	Correlations between serial advantage and discrete onset latency	19
4.6.3	Correlations between serial advantage and serial articulation times	20
4.7	Serial advantage vs. articulation: individual differences (include only children with negligible pause times)	21
4.7.1	Correlations between serial advantage and discrete articulation duration	22
4.7.2	Correlations between serial advantage and discrete onset latency	23
5	Serial articulation penalty	24
5.1	Graphs of articulation times per item	25
5.2	Graphs of rates (inverse-transformed times)	26
5.3	Graphs of means for each format	27
5.4	Statistical comparisons between formats (serial vs. discrete)	29
5.5	Correlations between serial articulation penalty and digit RAN	31
6	The one-overlap assumption	32
6.1	Graph of serial total time vs. discrete onset latency	32

6.2	Statistical comparison between serial pause+articulation and discrete onset latency	34
6.3	Correlations between serial pause+articulation and discrete onset latency	35

7	Session information	37
----------	----------------------------	-----------

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

1 Introduction

In this document we present analyses testing hypotheses involving the serial advantage and its relationship to individual response components.

Data quality considerations and descriptive statistics are presented in separate documents: “data check” and “descriptives”.

2 Read in data

Data from reading tests and discrete and serial naming for 298 children in Grades 1, 3, and 5 have been combined in a single wide-format data frame.

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

Variables are described in a separate document.

3 Differences between discrete and serial naming rates

Previous work has established that serial naming rates are higher than discrete naming rates. In other words, the total time to respond to a serial naming array is shorter than the sum of the times to respond to the same or comparable items individually in a discrete naming task (including both spoken and silent intervals in both cases).

3.1 Differences in total task time

The following set of graphs displays response time per item (including silent and spoken intervals) by grade and task content. The left (olive) box shows the discrete time, the middle (teal) box shows the raw serial time (i.e., total time divided by number of items), and the right (magenta) box shows adjusted serial time (i.e., adjusted time, after removing intervals with errors or fillers, divided by number of correct items). In addition, the faint gray lines show times for individual participants.

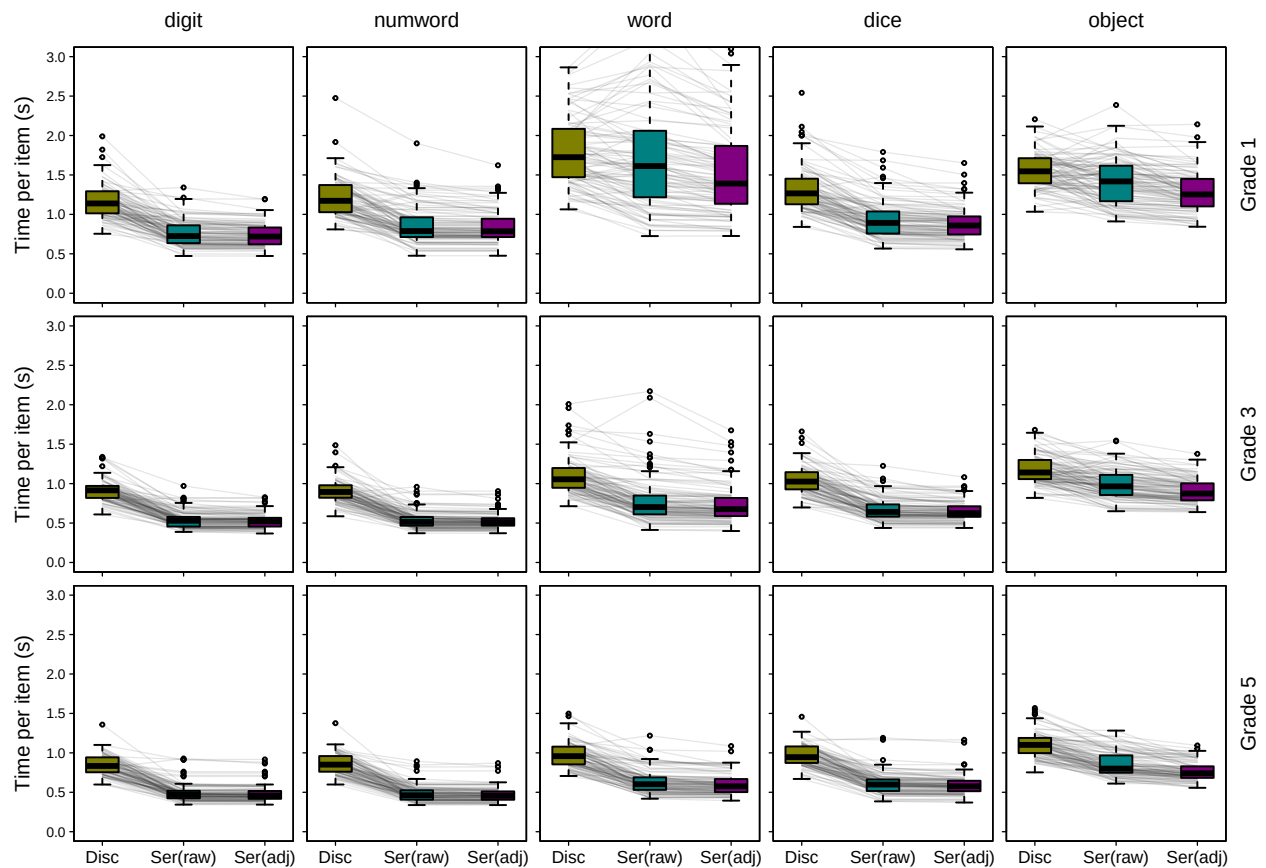
3.1.1 Graphs of total times per item

```
par(mfcol=c(3,5))
par(oma=c(2,3,2,2))
par(mai=c(0.1,0.1,0.01,0.01))
par(mgp=c(2,0.5,0))
par(tcl= -0.25)
for (tsk in tasks) {
  # calculate times per item
  rsdw[paste("RTsec" ,tsk,sep=".")] <- rsdw[paste("RT" ,tsk,sep=".")]/1000.0
```

```

rsdw[paste("itmtim",tsk,sep=".")] <- rsdw[paste("tottim",tsk,sep=".")]/36.0
for (grd in levels(rsdw$grade)) {
  plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0,3),yaxt="n",xaxt="n",xlab="",ylab="")
  if (grd=="G1") {
    mtext(tsk,side=3,line=1,xpd=NA)
  }
  if (grd=="G5") {
    axis(1,at=1:3,labels=c("Disc","Ser(raw)","Ser(adj)",cex.axis=1.25,xpd=NA)
  } else {
    axis(1,at=1:3,labels=F)
  }
  if (tsk==tasks[1]) {
    axis(2,at=0:6/2,las=1)
    mtext("Time per item (s)",2,line=2.5,xpd=NA)
  } else {
    axis(2,at=0:6/2,labels=F)
  }
  if (tsk==tasks[length(tasks)]) {
    mtext(paste("Grade",substr(grd,2,2)),side=4,line=1,xpd=NA)
  }
  for (ss in which(rsdw$grade==grd)) {
    lines(1:3,rsdw[ss,paste(c("RTsec","itmtim","itmadj"),tsk,sep=".")],col="#30303020")
  }
  boxplot(rsdw[rsdw$grade==grd,paste("RTsec",tsk,sep=".")],at=1,add=T,col="#808000",lwd=1.5,yaxt="n")
  boxplot(rsdw[rsdw$grade==grd,paste("itmtim",tsk,sep=".")],at=2,add=T,col="#008080",lwd=1.5,yaxt="n")
  boxplot(rsdw[rsdw$grade==grd,paste("itmadj",tsk,sep=".")],at=3,add=T,col="#800080",lwd=1.5,yaxt="n")
}
}

```



3.1.2 Graphs of rates (inverse-transformed times)

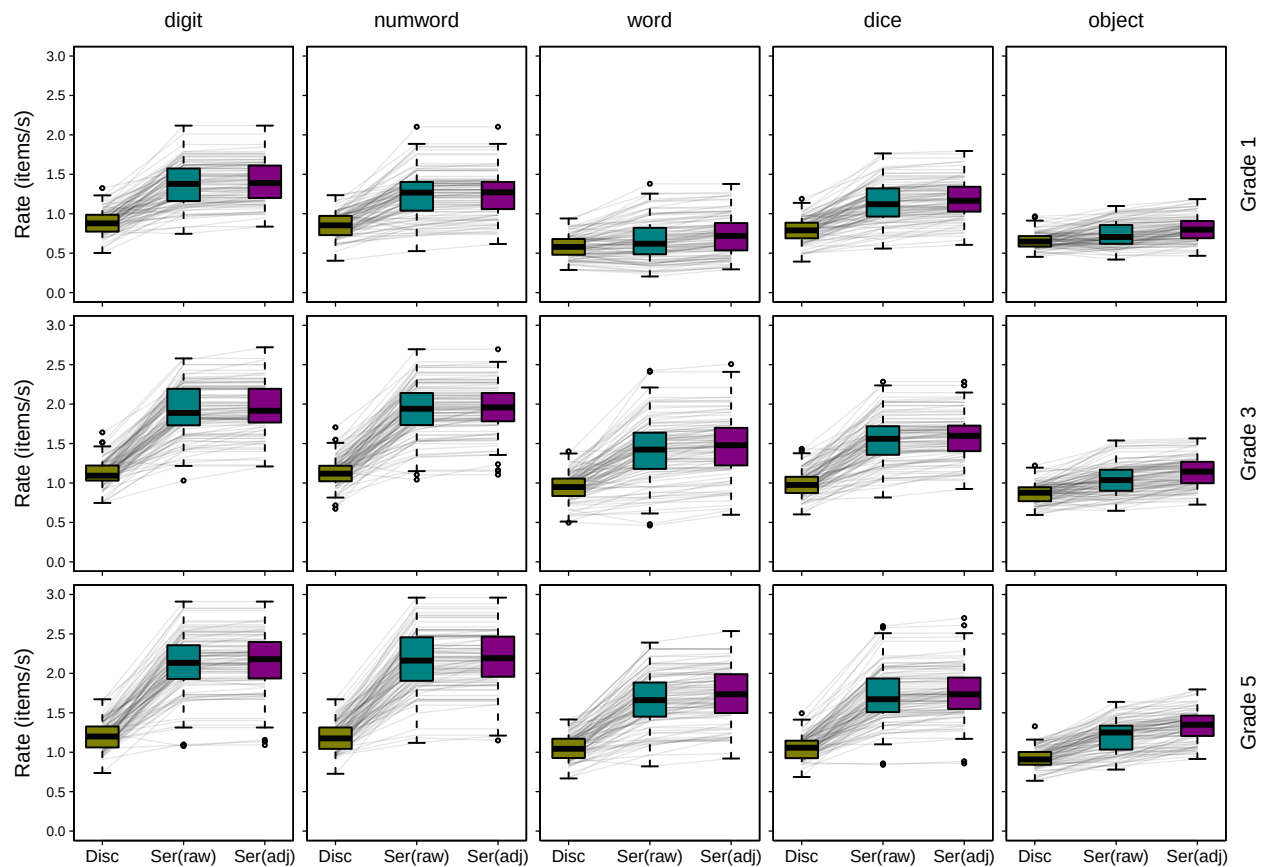
The following set of graphs display the exact same information in the rate scale (i.e., items per item), in which differences between grades are less pronounced due to more homogeneous variability.

```
par(mfcol=c(3,5))
par(oma=c(2,3,2,2))
par(mai=c(0.1,0.1,0.01,0.01))
par(mgp=c(2,0.5,0))
par(tcl= -0.25)
for (tsk in tasks) {
  for (grd in levels(rsdw$grade)) {
    plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0,3),yaxt="n",xaxt="n",xlab="",ylab="")
    if (grd=="G1") {
      mtext(tsk,side=3,line=1,xpd=NA)
    }
    if (grd=="G5") {
      axis(1,at=1:3,labels=c("Disc","Ser(raw)","Ser(adj)"),cex.axis=1.25,xpd=NA)
    } else {
      axis(1,at=1:3,labels=F)
    }
    if (tsk==tasks[1]) {
      axis(2,at=0:6/2,las=1)
      mtext("Rate (items/s)",2,line=2.5,xpd=NA)
    } else {
      axis(2,at=0:6/2,labels=F)
    }
  }
}
```

```

}
if (tsk==tasks[length(tasks)]) {
  mtext(paste("Grade",substr(grd,2,2)),side=4,line=1,xpd=NA)
}
for (ss in which(rsdw$grade==grd)) {
  lines(1:3,rsdw[ss,paste(c("invRT","invraw","invadj"),tsk,sep=".")],col="#30303020")
}
boxplot(rsdw[rsdw$grade==grd,paste("invRT",tsk,sep=".")],at=1,add=T,col="#808000",lwd=1.5,yaxt="n")
boxplot(rsdw[rsdw$grade==grd,paste("invraw",tsk,sep=".")],at=2,add=T,col="#008080",lwd=1.5,yaxt="n")
boxplot(rsdw[rsdw$grade==grd,paste("invadj",tsk,sep=".")],at=3,add=T,col="#800080",lwd=1.5,yaxt="n")
}
}

```



These graphs show that there is a sizable serial advantage overall, as expected. More specifically, there are consistently higher rates (shorter times) in serial than discrete tasks. In most cases there is no difference between raw and adjusted serial times/rates.

Some interesting exceptions from the general patterns are:

- * For words, participants with the lowest performance (esp. in G1) do not show a serial advantage; however, participants with the highest performance (esp. not in G1) show a very large serial advantage.
- * For objects, serial advantage is the smallest, indeed disproportionally small even if we consider the generally low baseline (i.e., discrete) performance.
- * For objects, the serial advantage is greater when only correct responses are considered (i.e., using adjusted rates), whereas there is no obvious difference between adjusted and unadjusted times/rates for other types of content.

4 The serial advantage

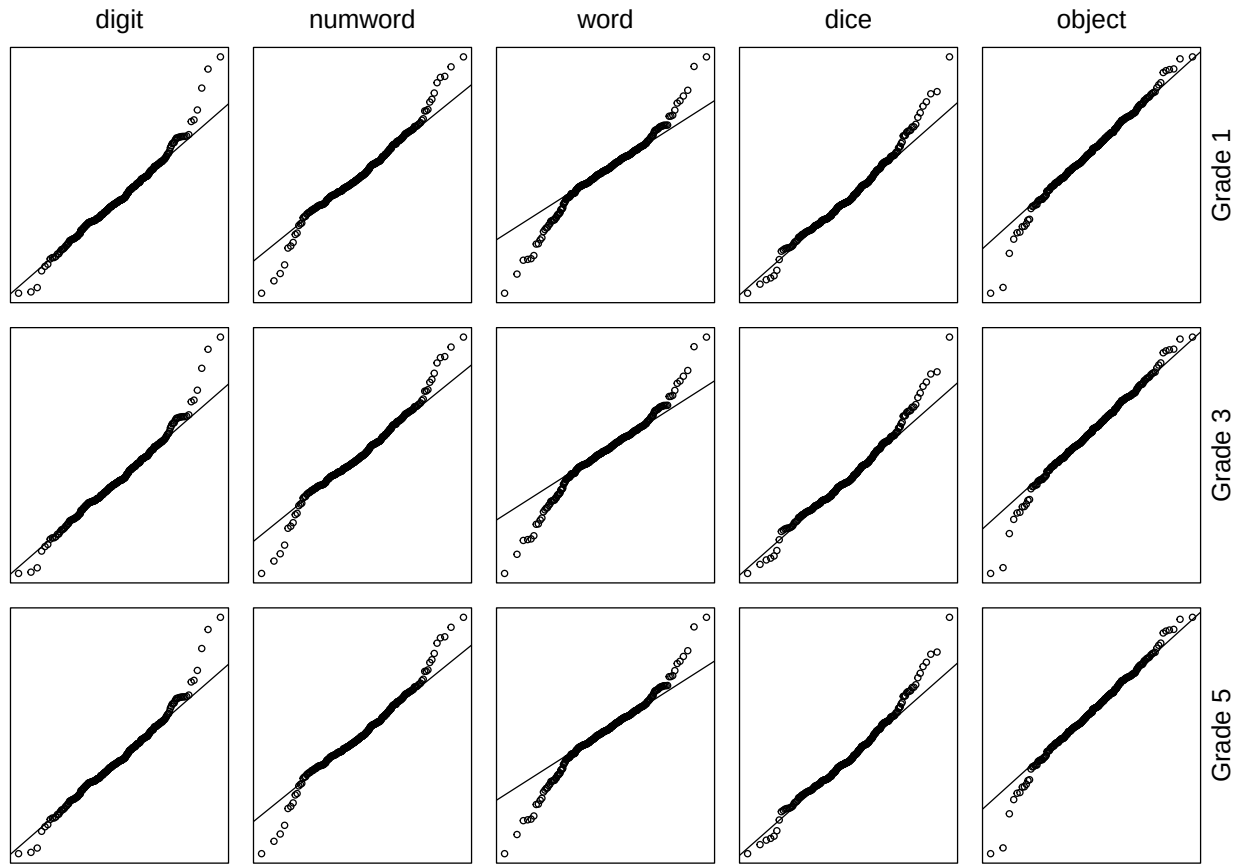
4.1 Calculation and distributions

We first calculate the serial advantage (per item) for each task and child by subtracting the total serial time (per correct item) from the total discrete time (per correct item).

```
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=".")]  
}
```

Serial advantage turns out to be reasonably normally distributed for the most part, so we can use the mean as an index of central tendency.

```
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)  
#seradv$logseradv <- ifelse(seradv$rawseradv>0, log(seradv$rawseradv), NA)  
#seradv$inuseradv <- ifelse(seradv$rawseradv>0, 1.0/seradv$rawseradv, NA)  
  
ids <- levels(seradv$sID)  
  
par(mfrow=c(3,5))  
par(oma=c(0,0,2,2))  
par(mai=c(0.1,0.1,0.1,0.1))  
par(mgp=c(2,0.5,0))  
par(tcl= -0.3)  
  
for (grd in levels(seradv$grade)) {  
  for (tsk in tasks) {  
    gdat <- rsdw[,paste("seradv",tsk,sep=".")]  
    qqnorm(gdat,main="",xlab="",ylab="",xaxt="n",yaxt="n")  
    qqline(gdat)  
  
    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) }  
  }  
}
```



4.2 Serial advantage predictions

Processing considerations lead to the following predictions: Under the assumptions that

(a) there are only two processing streams that do not interfere with each other (“processing” vs. “articulation”) and

(b) processing is overlapped with maximum efficiency,

then the serial advantage (i.e., the difference between total discrete time and total serial time) will tend to:
 * processing time, if task performance is articulation-limited (i.e., articulation takes longer than processing, in which case silent intervals/pause times tend to zero and articulation is nearly continuous).

* articulation time, if task performance is processing-limited (i.e., processing takes longer than articulation, in which case there are nonnegligible silent intervals between articulation intervals in the response).

In the present data, silent intervals are consistently observed (even as they become quite small for the most automatized tasks by Grade 5), so the relevant prediction is the latter.

4.3 Serial advantage vs. discrete silent and spoken intervals

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).

4.3.1 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))
```

```

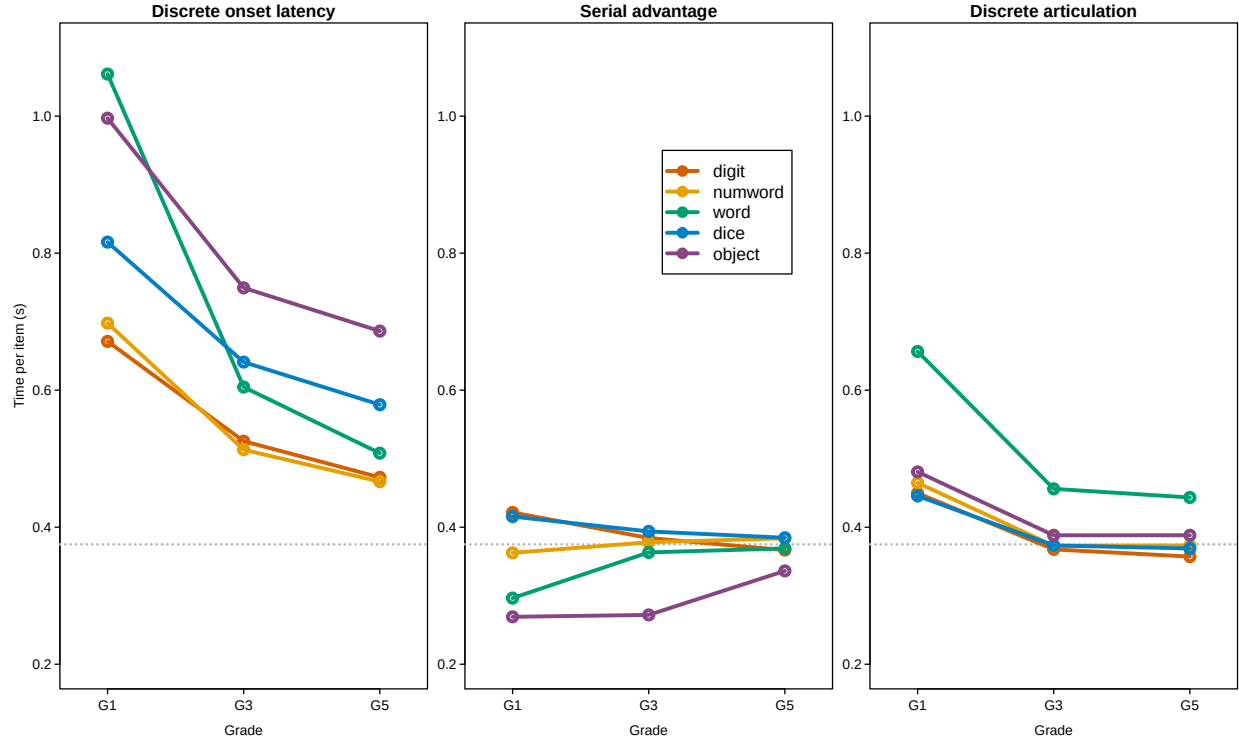
par(mgp=c(2,0.5,0))
par(tcl= -0.3)

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(0.2,1.1),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(0.2,1.1),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(2.1,0.95,tasks,col=tcols,pch=21,lty=1,lwd=3,cex=1.25)

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(0.2,1.1),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.

4.3.2 Statistical comparison

We cannot statistically compare these means directly because the two quantities are not similarly distributed (and articulation duration, in particular, is highly skewed). However, we can apply a (nonparametric) Wilcoxon signed rank test to compare the corresponding medians. The table below displays each pair of means and corresponding statistic (W) and significance (p) for the statistical comparison.

```
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.

4.4 Serial advantage vs. serial silent and spoken intervals

4.4.1 Graphical comparison

The following set of panels follow the exact same idea (and axes), but using comparison data from the serial tasks instead. That is, instead of comparing the mean serial advantage to discrete onset latency and articulation duration, we compare it to serial silent and speech intervals, namely pause and articulation times, respectively.

```

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)

serons <- exp(aggregate( as.formula( paste("cbind(", paste(paste("logpautE",tasks,sep="."),collapse=","),
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.2,1.1),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("logpautE",tasks[itsk],sep=".")

```

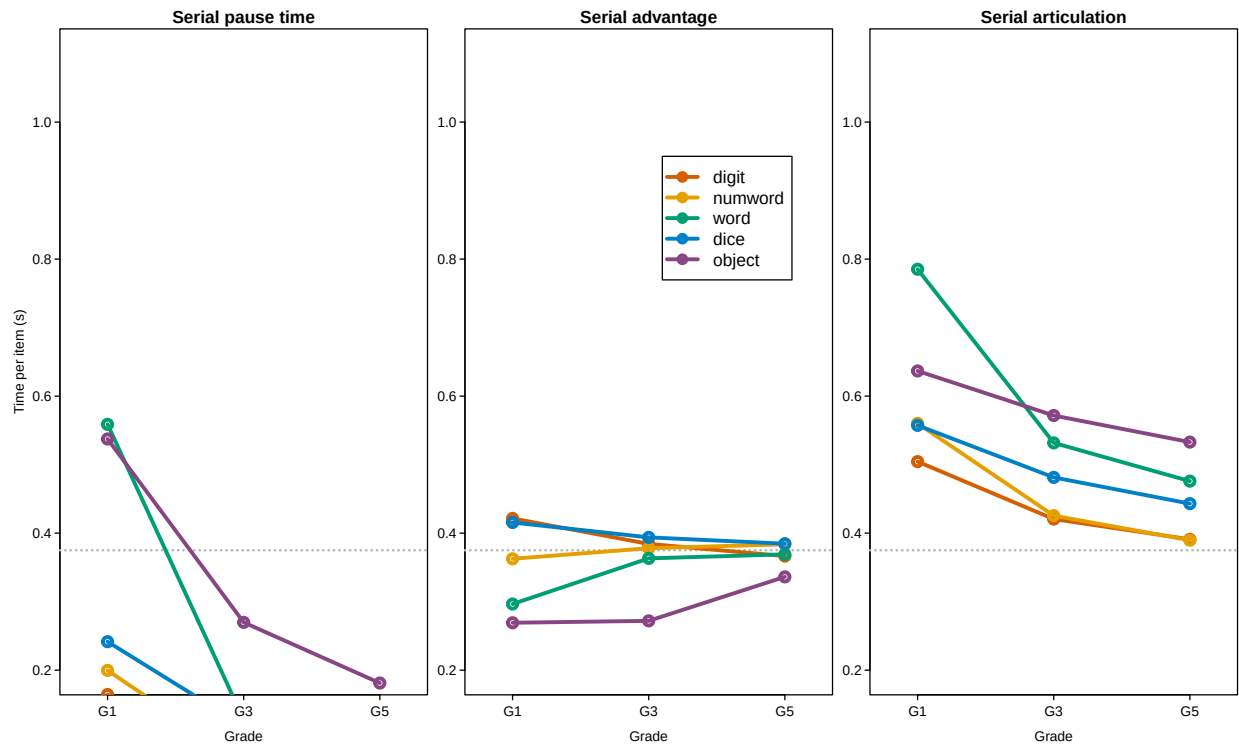
```

    lines(1:3,serons[,tsk],col=tcols[itsk],lwd=3)
    points(1:3,serons[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
  }

seradvm <- aggregate( as.formula( paste("cbind(", paste(paste("seradv",tasks,sep="."),collapse=","), "
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.2,1.1),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,seradvm[,tsk],col=tcols[itsk],lwd=3)
  points(1:3,seradvm[,tsk],col=tcols[itsk],lwd=3,pch=21,cex=1.5)
}
legend(2.1,0.95,tasks,col=tcols,pch=21,lty=1,lwd=3,cex=1.25)

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=c(0.2,1.1),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("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)
}

```



It is clear that only digits and number words in Grade 5 come close to the serial advantage approaching articulation duration. Overall, and in contrast to the pattern using discrete data, this pattern is not consistent with expectations based on the aforementioned hypothesis. We submit that this is because the durations of silent and spoken intervals in serial tasks are not pure indicators of the time needed for processing and

articulation, respectively, in contrast to discrete tasks.

4.4.2 Statistical comparison

For completeness we also perform the corresponding statistical comparison.

```
seradv_serart_tab <- numeric()
for (grd in levels(rsdw$grade)) {
  glines <- numeric()
  for (tsk in tasks) {
    #print(paste(grd,tsk))
    seradvgt <- rsdw[rsdw$grade==grd,paste("seradv",tsk,sep=".")]
    serartgt <- exp(rsdw[rsdw$grade==grd,paste("logarttE",tsk,sep=".")])
    wilcox.test(seradvgt,serartgt,paired=T) -> wt
    line <- c(round(mean(seradvgt,na.rm=T),3), round(mean(serartgt,na.rm=T),3), wt$statistic, wt$p.value)
    glines <- rbind(glines,line)
  }
  colnames(glines) <- paste(c("seradv","serart","W","p"),grd,sep=".")
  seradv_serart_tab <- cbind(seradv_serart_tab,glines)
}
rownames(seradv_serart_tab) <- tasks

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

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

##	seradv.G1	serart.G1	W.G1	p.G1	seradv.G3	serart.G3	W.G3	p.G3	seradv.G5	serart.G5	W.G5	p.G5
## digit	0.421	0.512	1338	0	0.384	0.421	1698	0.004	0.365	0.394	2059	0.254
## numword	0.371	0.568	507	0	0.374	0.425	1523	0.000	0.384	0.393	2174	0.467
## word	0.302	0.810	38	0	0.362	0.541	459	0.000	0.366	0.481	645	0.000
## dice	0.421	0.564	758	0	0.393	0.483	1104	0.000	0.380	0.446	1131	0.000
## object	0.270	0.646	123	0	0.273	0.575	19	0.000	0.338	0.537	28	0.000

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

##	seradv.G1	serart.G1	W.G1	p.G1	seradv.G3	serart.G3	W.G3	p.G3	seradv.G5	serart.G5	W.G5	p.G5
## digit	0.421	0.512	1338	0.001	0.384	0.421	1698	0.013	0.365	0.394	2059	0.508
## numword	0.371	0.568	507	0.000	0.374	0.425	1523	0.001	0.384	0.393	2174	0.508
## word	0.302	0.810	38	0.000	0.362	0.541	459	0.000	0.366	0.481	645	0.000
## dice	0.421	0.564	758	0.000	0.393	0.483	1104	0.000	0.380	0.446	1131	0.000
## object	0.270	0.646	123	0.000	0.273	0.575	19	0.000	0.338	0.537	28	0.000

Here, only digits and number words in Grade 5 fail to reach statistical significance.

4.5 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.

4.5.1 Correlations between serial advantage and discrete articulation duration

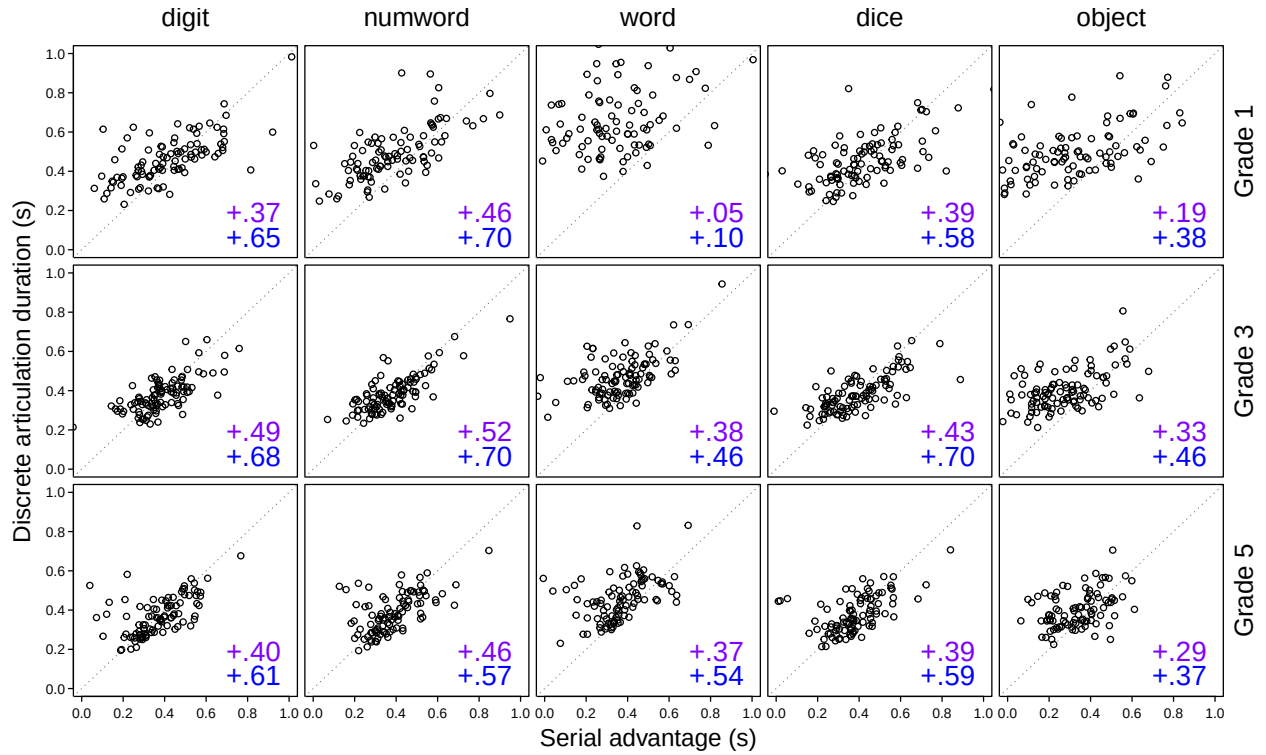
On each scatterplot, the corresponding Spearman's ρ is displayed in blue, and the unstandardized β regression coefficient is displayed in violet/purple.

```
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(1,0.05,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(1,0.175,lmbetastr,cex=2,pos=2,col="#8000FF")
    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.

4.5.2 Correlations between serial advantage and discrete onset latency

Next we plot the same graphs of serial advantage but this time against discrete onset latency instead of articulation.

```
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 <- "onsRTsec"
v2s <- "seradv"

for (grd in levels(seradv$grade)) {
  for (tsk in tasks) {
    rsdw[,paste(v1s,tsk,sep=".")] <- exp(rsdw[,paste("logonsRT",tsk,sep=".")])/1000

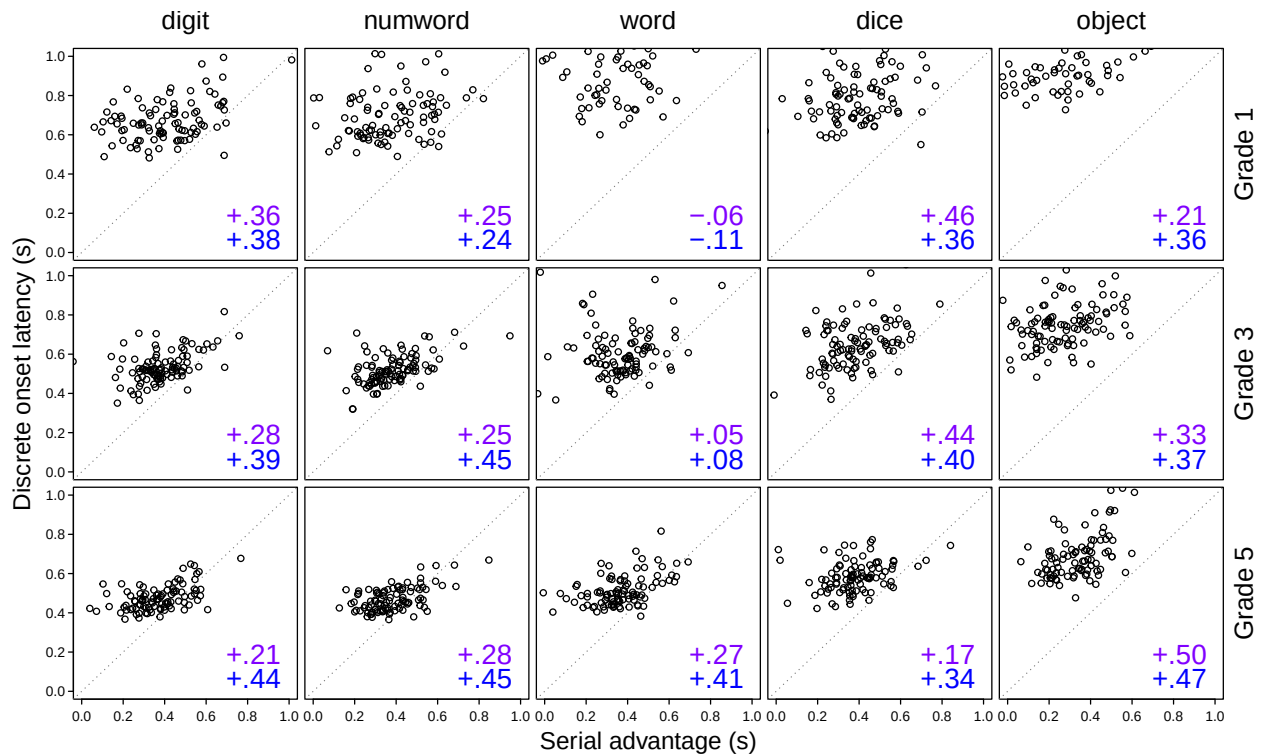
    plot(as.formula(paste(paste(v1s,tsk,sep="."),paste(v2s,tsk,sep="."),sep="~")),
         rsdw,subset= grade==grd,
         xlim=c(0,1),ylim=c(0,1),xaxt="n",yaxt="n")
    if (grd=="G5") { axis(1) }
    if (tsk==tasks[1]) { axis(2,las=1) }
    abline(0,1,lty=3,col="grey50")
    cstr <- signs(
      cor(rsdw[rsdw$grade==grd,paste(v1s,tsk,sep=".")],
```

```

    rsdw[rsdw$grade==grd,paste(v2s,tsk,sep=".")],
    method="spearman",use="pairwise"),
    accuracy = .01, trim_leading_zeros = TRUE, add_plusses=TRUE)
text(1,0.05,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(1,0.175,lmbetastr,cex=2,pos=2,col="#8000FF")

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 moderate; certainly lower than with discrete articulation, and clearly farther from the diagonal.

4.5.3 Correlations between serial advantage and serial articulation times

Next we plot the same kind of graphs but using serial instead of discrete articulation, for comparison:

```

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 <- "itmarttE"
v2s <- "seradv"

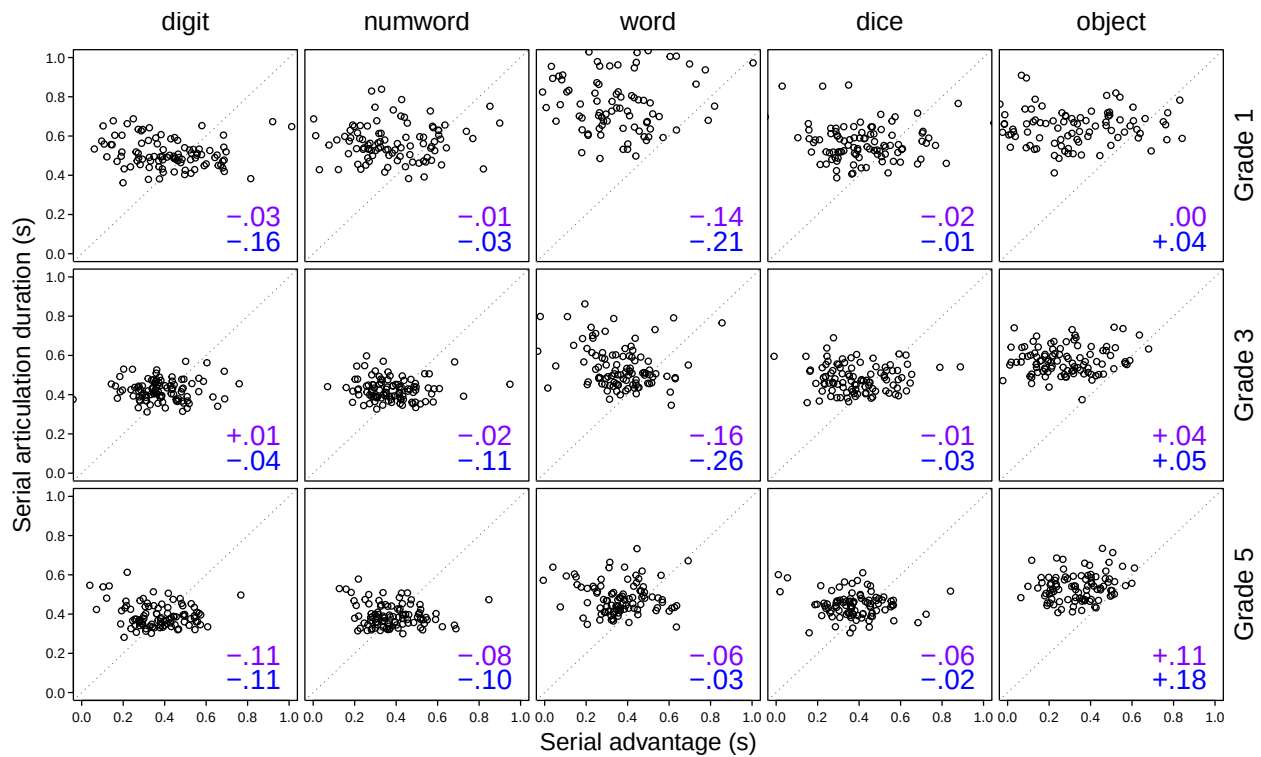
for (grd in levels(seradv$grade)) {
  for (tsk in tasks) {

    plot(as.formula(paste(paste(v1s,tsk,sep="."),paste(v2s,tsk,sep="."),sep="~")),
         rsdw,subset= grade==grd,
         xlim=c(0,1),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(1,0.05,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(1,0.175,lmbetastr,cex=2,pos=2,col="#8000FF")

    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("Serial articulation duration (s)", 2, outer=T, cex=1.2, line=2.2)

```



These are entirely uncorrelated.

4.6 Serial advantage vs. articulation: individual differences (exclude children with negligible pause times)

Now let us look at the same graphs after excluding participants approaching continuous articulation, that is, silent intervals tending to zero. If we allow for a breath at every line, mean adjusted pause times less than 0.05 s would be approaching the limit.

```
for (tsk in tasks) {
  rsdw[,paste("artic2",tsk,sep=".")] <- ifelse ( rsdw[,paste("itmpaute",tsk,sep=".")] < 0.05, NA, rsdw[
  rsdw[,paste("itmarttE2",tsk,sep=".")] <- ifelse ( rsdw[,paste("itmpaute",tsk,sep=".")] < 0.05, NA, rsdw[
  rsdw[,paste("onsRT2sec",tsk,sep=".")] <- ifelse ( rsdw[,paste("itmpaute",tsk,sep=".")] < 0.05, NA, rsdw[
}
```

4.6.1 Correlations between serial advantage and discrete articulation duration

```
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 <- "articsec2"
v2s <- "seradv"

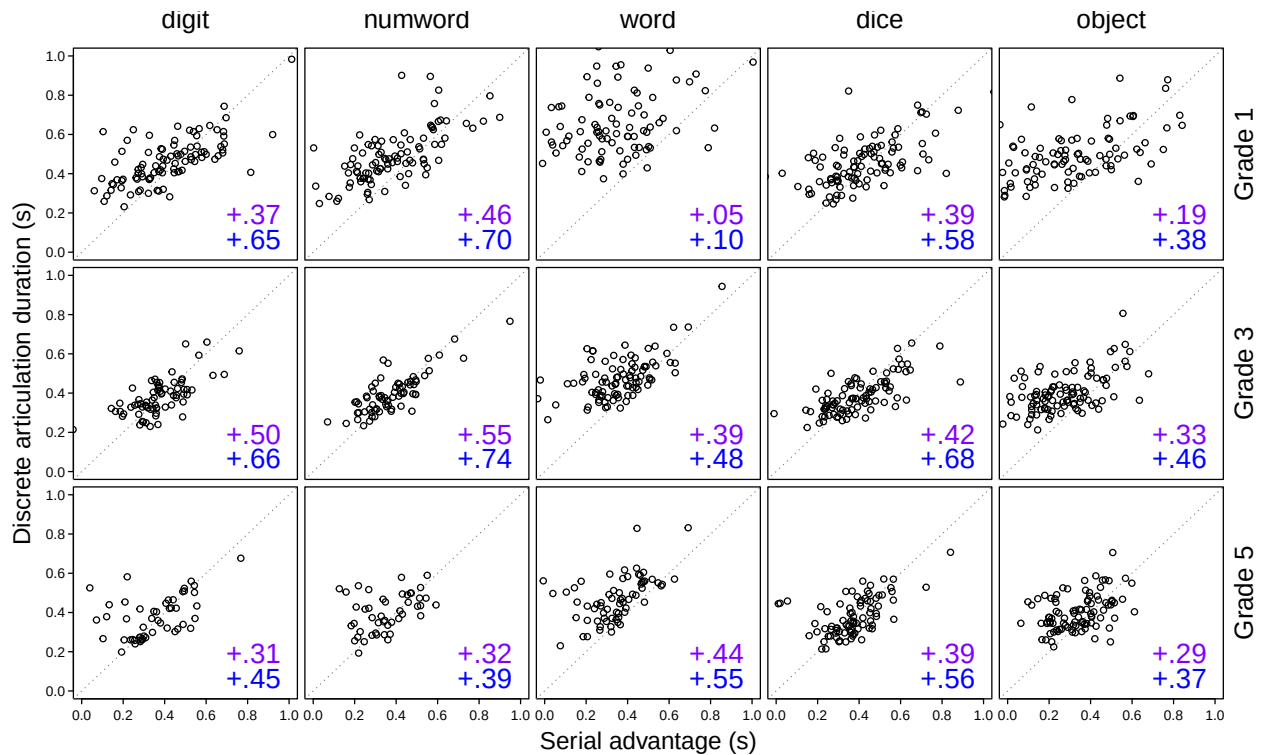
for (grd in levels(seradv$grade)) {
  for (tsk in tasks) {
    rsdw[,paste(v1s,tsk,sep=".")] <- rsdw[,paste("artic2",tsk,sep=".")]/1000
```

```

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(1,0.05,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(1,0.175,lmbetastr,cex=2,pos=2,col="#8000FF")

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)

```



The pattern remains essentially unchanged, confirming the prediction.

4.6.2 Correlations between serial advantage and discrete onset latency

The same kind of graphs but using discrete onset latency instead of articulation, for comparison:

```
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)

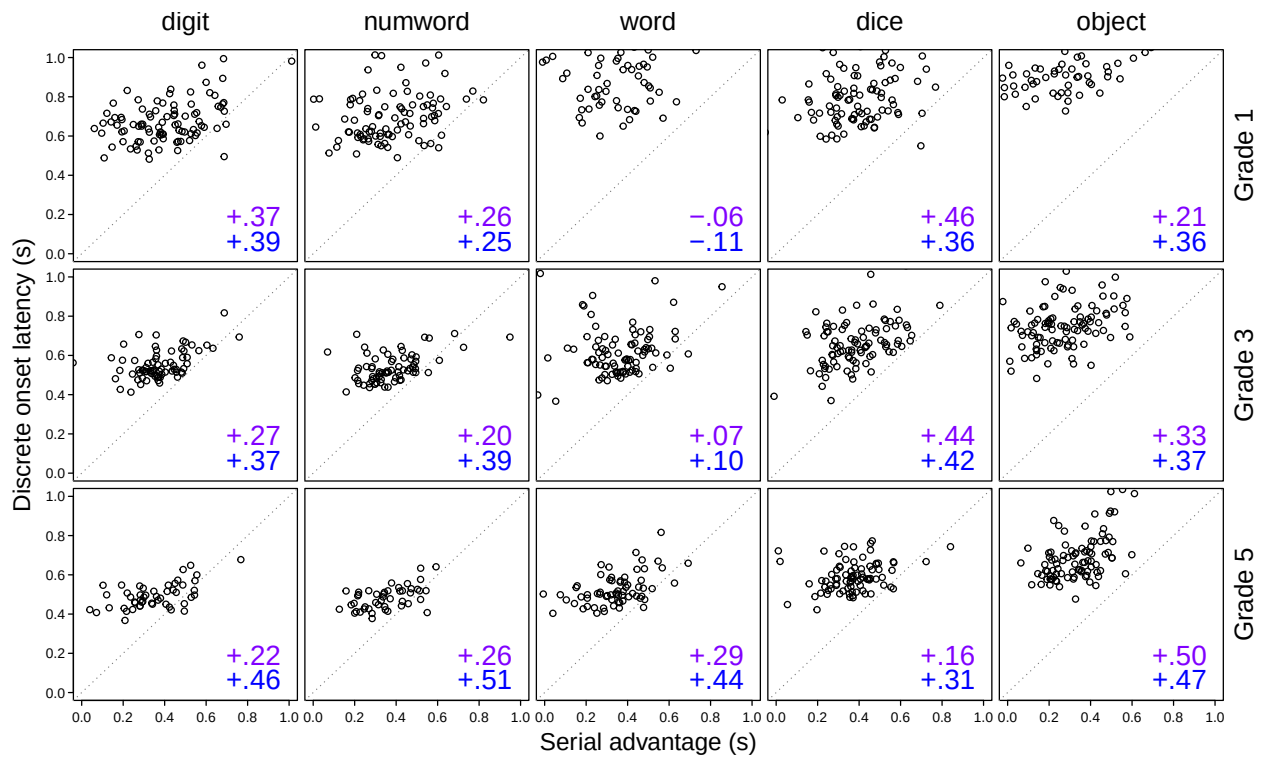
v1s <- "onsRT2sec"
v2s <- "seradv"

for (grd in levels(seradv$grade)) {
  for (tsk in tasks) {

    plot(as.formula(paste(paste(v1s,tsk,sep=". "),paste(v2s,tsk,sep=". "),sep="~")),
         rsdw,subset= grade==grd,
         xlim=c(0,1),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(1,0.05,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(1,0.175,lmbetastr,cex=2,pos=2,col="#8000FF")

    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)
```



As with the full sample, these correlations are smaller, and the points clearly farther from the diagonal, compared to the those with discrete articulation.

4.6.3 Correlations between serial advantage and serial articulation times

Again, the same kind of graphs but using serial instead of discrete articulation, for comparison:

```
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 <- "itmarttE2"
v2s <- "seradv"

for (grd in levels(seradv$grade)) {
  for (tsk in tasks) {

    plot(as.formula(paste(paste(v1s,tsk,sep="."),paste(v2s,tsk,sep="."),sep="~")),
         rsdw,subset= grade==grd,
         xlim=c(0,1),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)
```

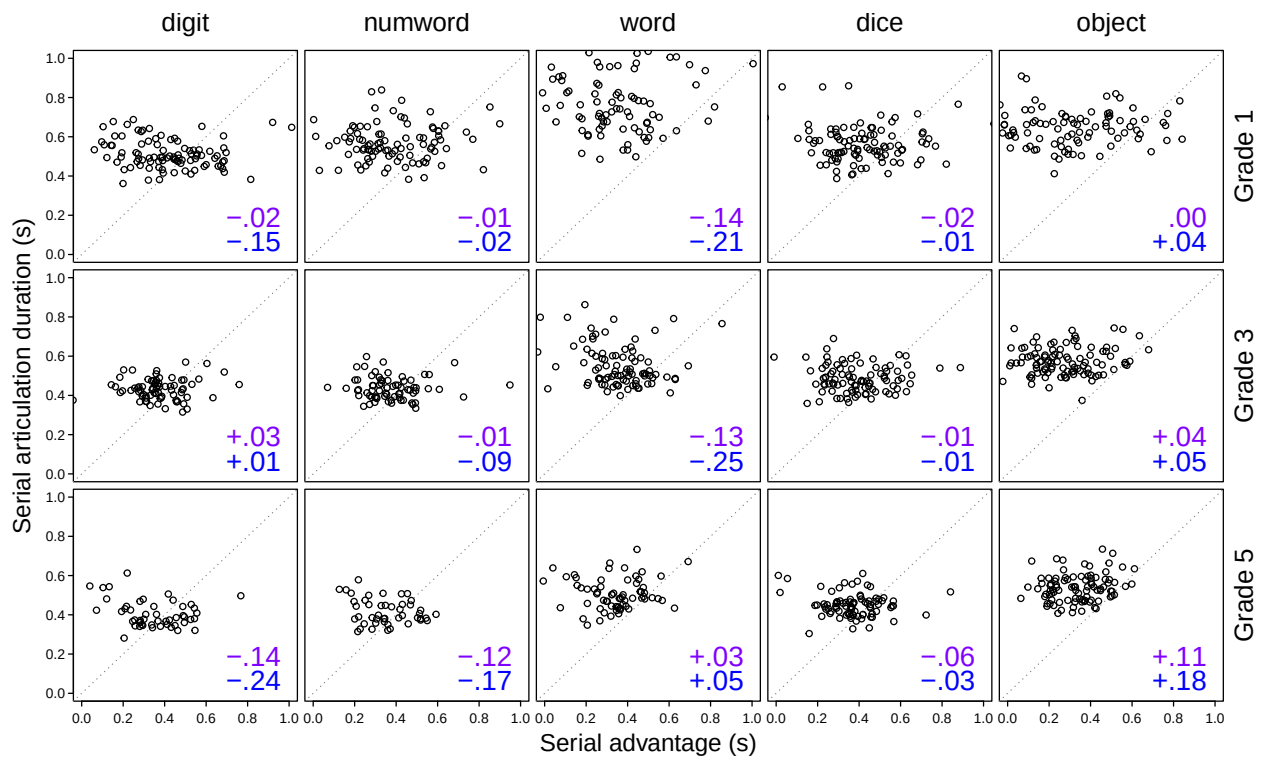
```

text(1,0.05,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(1,0.175,lmbetastr,cex=2,pos=2,col="#8000FF")

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("Serial articulation duration (s)", 2, outer=T, cex=1.2, line=2.2)

```



And these are, again, uncorrelated, for the most relevant subset of children.

4.7 Serial advantage vs. articulation: individual differences (include only children with negligible pause times)

Finally, let us look at the relevant graphs including only participants approaching continuous articulation (those that were excluded in the preceding graphs).

```

for (tsk in tasks) {
  rsdw[,paste("artic3",tsk,sep=".")] <- ifelse ( rsdw[,paste("itmpautE",tsk,sep=".")] >= 0.05, NA, rsdw[,paste("artic3",tsk,sep=".")] )
  rsdw[,paste("itmarttE3",tsk,sep=".")] <- ifelse ( rsdw[,paste("itmpautE",tsk,sep=".")] >= 0.05, NA, rsdw[,paste("itmarttE3",tsk,sep=".")] )
  rsdw[,paste("logonsRT3",tsk,sep=".")] <- ifelse ( rsdw[,paste("itmpautE",tsk,sep=".")] >= 0.05, NA, rsdw[,paste("logonsRT3",tsk,sep=".")] )
}

```

These groups are small (often empty), so the following graphs should be viewed with reservation.

```
apply(rsdw[,paste("artic3",tasks,sep=".")],2,function(x){tapply(x,rsdw$grade,function(y){sum(!is.na(y))})})
```

```
##      artic3.digit artic3.numword artic3.word artic3.dice artic3.object
## G1              1              1           0           0           0
## G3              28             30          11           8           0
## G5              46             51          29          12           0
```

4.7.1 Correlations between serial advantage and discrete articulation duration

```
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)

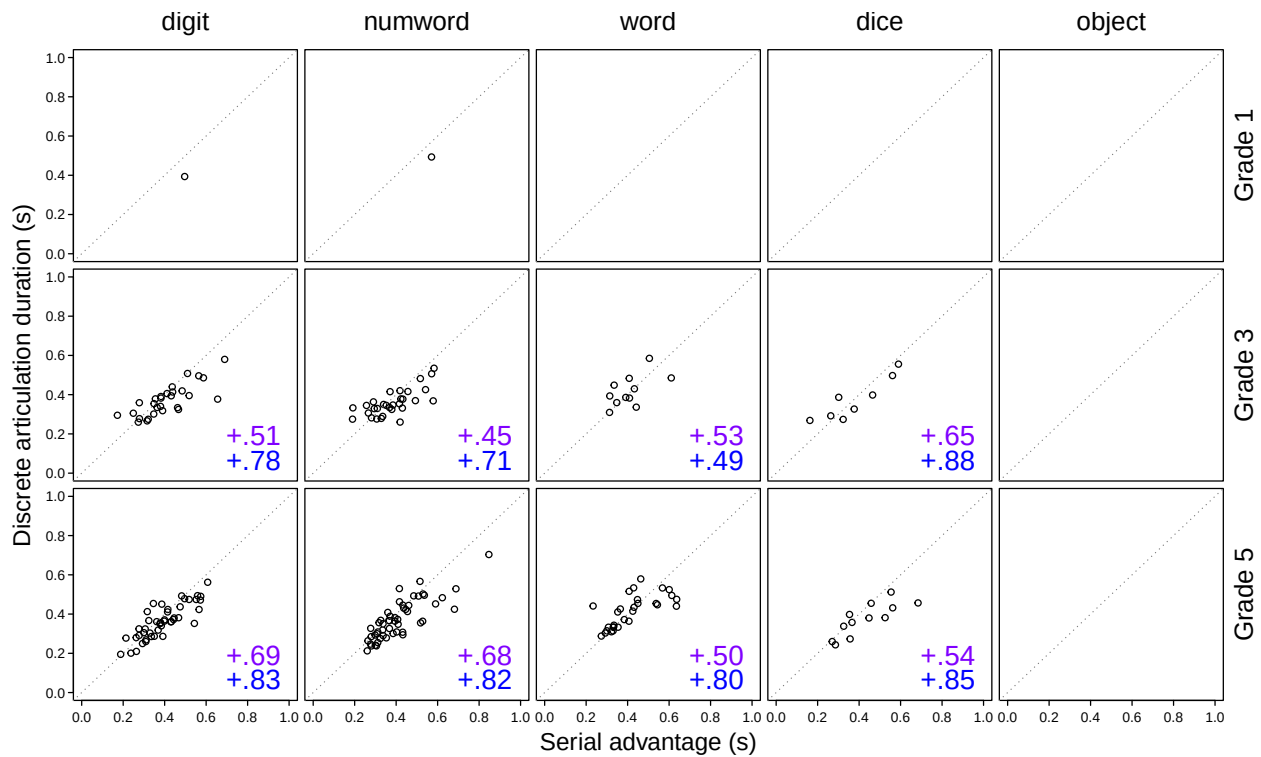
v1s <- "articsec3"
v2s <- "seradv"

for (grd in levels(seradv$grade)) {
  for (tsk in tasks) {
    rsdw[,paste(v1s,tsk,sep=".")] <- rsdw[,paste("artic3",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 (sum(!is.na(rsdw[rsdw$grade==grd,paste(v1s,tsk,sep=".)]))>1) {
      cstr <- signs(
        cor(rsdw[rsdw$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(1,0.05,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(1,0.175,lmbetastr,cex=2,pos=2,col="#8000FF")
      )
    }

    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)
```



These correlations are quite strong indeed.

4.7.2 Correlations between serial advantage and discrete onset latency

Again, the same kind of graphs but using discrete onset latency instead of articulation, for comparison:

```
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 <- "onsRTsec3"
v2s <- "seradv"

for (grd in levels(seradv$grade)) {
  for (tsk in tasks) {
    rsdw[,paste(v1s,tsk,sep=".")] <- exp(rsdw[,paste("logonsRT3",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 (sum(!is.na(rsdw[rsdw$grade==grd,paste(v1s,tsk,sep=".)]))>1) {
      cstr <- signs(
        cor(rsdw[rsdw$grade==grd,paste(v1s,tsk,sep=".)],
            rsdw[rsdw$grade==grd,paste(v2s,tsk,sep=".)],
            method="spearman",use="pairwise"),
```

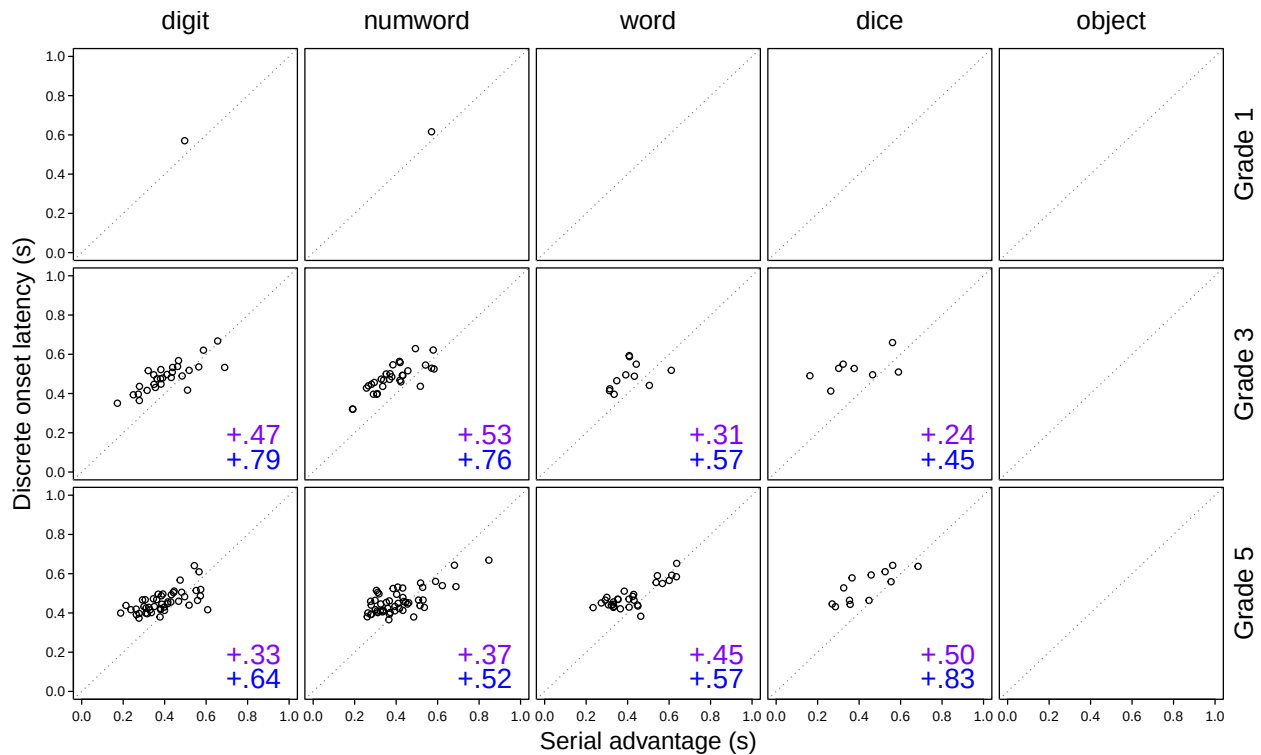
```

    accuracy = .01, trim_leading_zeros = TRUE, add_plusses=TRUE)
text(1,0.05,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(1,0.175,lmbetastr,cex=2,pos=2,col="#8000FF")
}

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)

```



As predicted by the overlap hypothesis, and in contrast to the children with nonnegligible pauses, these are moderately to strongly correlated and the points are closer to the diagonal.

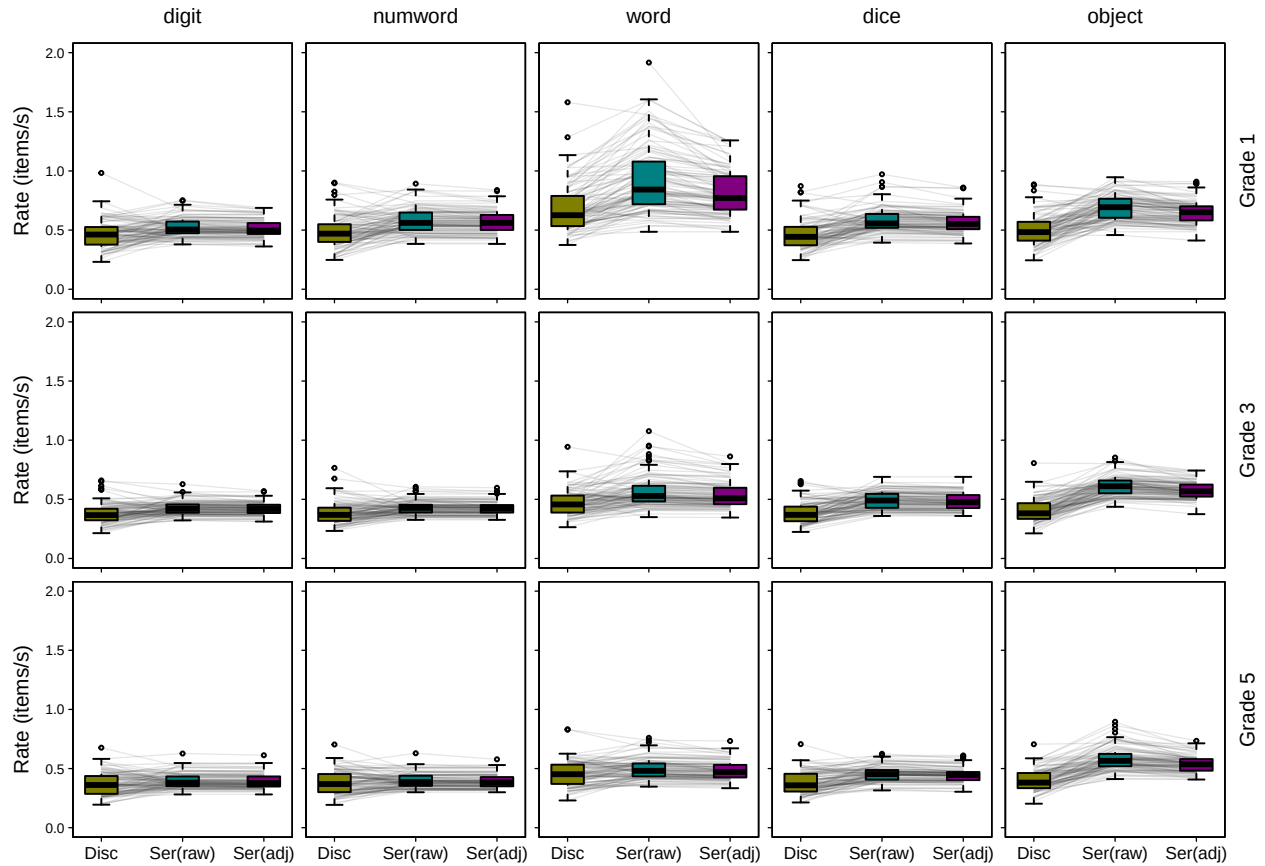
5 Serial articulation penalty

While designing this study we assumed that items would require the same time to be uttered regardless of format, therefore the duration of spoken intervals per item should not depend on task format. That is, we did not anticipate a difference in articulation rates between formats.

5.1 Graphs of articulation times per item

The following set of graphs displays articulation time by grade and task content. Graphical parameters are the same as above.

```
par(mfcol=c(3,5))
par(oma=c(2,3,2,2))
par(mai=c(0.1,0.1,0.01,0.01))
par(mgp=c(2,0.5,0))
par(tcl= -0.25)
for (tsk in tasks) {
  for (grd in levels(rsdw$grade)) {
    ssd <- rsdw[rsdw$grade==grd,paste(c("invaritic","arttot","arttotE","totcor"),tsk,sep=".")]
    ssd[paste("dart" ,tsk,sep=".")] <- 1.0/ssd[paste("invaritic",tsk,sep=".")]
    ssd[paste("sart" ,tsk,sep=".")] <- ssd[paste("arttot",tsk,sep=".)"]/36.0
    ssd[paste("sartE",tsk,sep=".")] <- ssd[paste("arttotE",tsk,sep=".)] /
      ssd[paste("totcor",tsk,sep=".)]
    plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0,2),yaxt="n",xaxt="n",xlab="",ylab="")
    if (grd=="G1") {
      mtext(tsk,side=3,line=1,xpd=NA)
    }
    if (grd=="G5") {
      axis(1,at=1:3,labels=c("Disc","Ser(raw)","Ser(adj)"),cex.axis=1.25,xpd=NA)
    } else {
      axis(1,at=1:3,labels=F)
    }
    if (tsk==tasks[1]) {
      axis(2,at=0:4/2,las=1)
      mtext("Rate (items/s)",2,line=2.5,xpd=NA)
    } else {
      axis(2,at=0:4/2,labels=F)
    }
    if (tsk==tasks[length(tasks)]) {
      mtext(paste("Grade",substr(grd,2,2)),side=4,line=1,xpd=NA)
    }
    for (ss in 1:nrow(rsdw)) {
      lines(1:3,ssd[ss,paste(c("dart","sart","sartE"),tsk,sep=".)],col="#30303020")
    }
    boxplot(ssd[,paste("dart" ,tsk,sep=".)],at=1,add=T,col="#808000",lwd=1.5,yaxt="n")
    boxplot(ssd[,paste("sart",tsk,sep=".)],at=2,add=T,col="#008080",lwd=1.5,yaxt="n")
    boxplot(ssd[,paste("sartE",tsk,sep=".)],at=3,add=T,col="#800080",lwd=1.5,yaxt="n")
  }
}
```



5.2 Graphs of rates (inverse-transformed times)

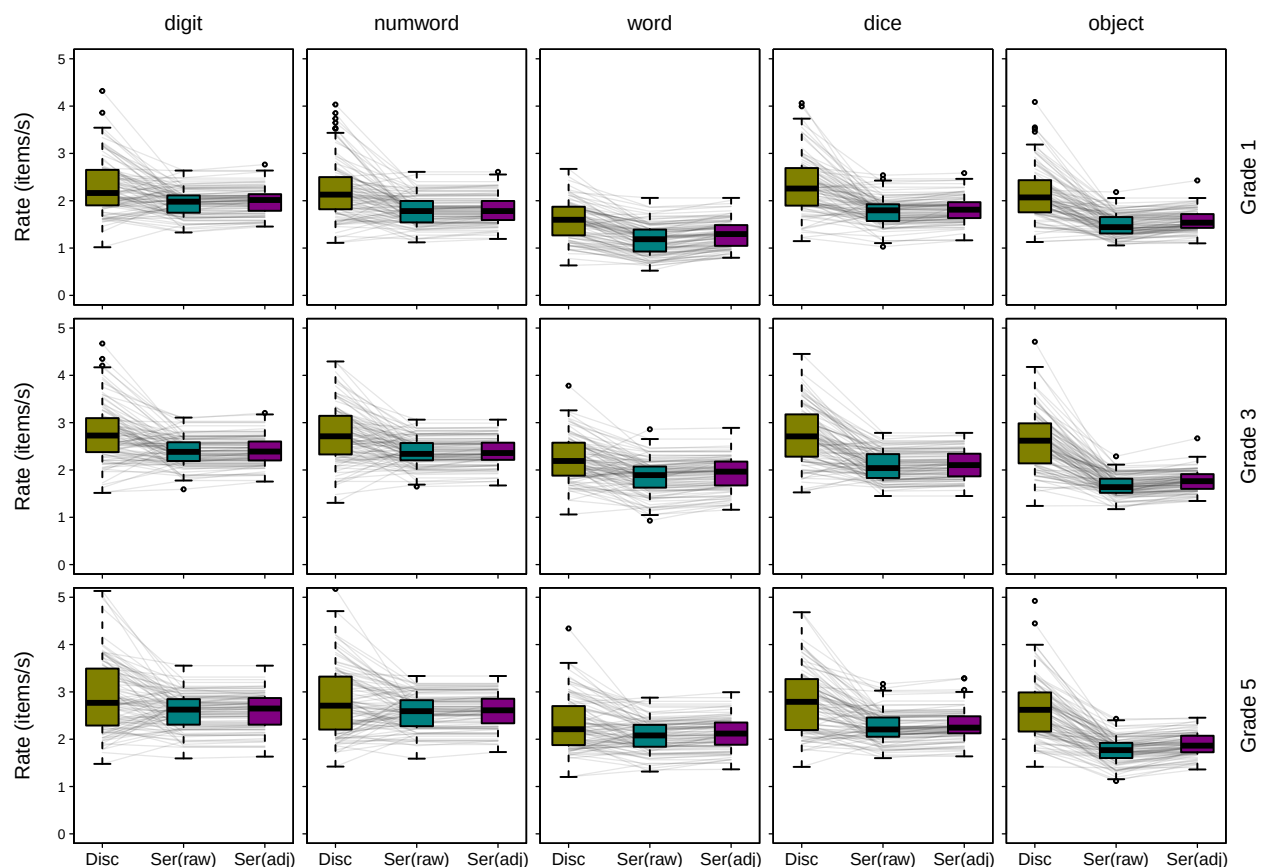
The following set of graphs displays the same information as rate (items per second).

```
par(mfcol=c(3,5))
par(oma=c(2,3,2,2))
par(mai=c(0.1,0.1,0.01,0.01))
par(mgp=c(2,0.5,0))
par(tcl= -0.25)
for (tsk in tasks) {
  for (grd in levels(rsdw$grade)) {
    ssd <- rsdw[rsdw$grade==grd,paste(c("invtartic","arttot","arttotE","totcor"),tsk,sep=".")]
    ssd[paste("invsart",tsk,sep=".")] <- 36.0/ssd[paste("arttot",tsk,sep=".")]
    ssd[paste("invsartE",tsk,sep=".")] <- ssd[paste("totcor",tsk,sep=".)"]/ssd[paste("arttotE",tsk,sep=
    plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0,5),yaxt="n",xaxt="n",xlab="",ylab="")
    if (grd=="G1") {
      mtext(tsk,side=3,line=1,xpd=NA)
    }
    if (grd=="G5") {
      axis(1,at=1:3,labels=c("Disc","Ser(raw)","Ser(adj)"),cex.axis=1.25,xpd=NA)
    } else {
      axis(1,at=1:3,labels=F)
    }
    if (tsk==tasks[1]) {
      axis(2,at=0:5,las=1)
      mtext("Rate (items/s)",2,line=2.5,xpd=NA)
    }
  }
}
```

```

} else {
  axis(2,at=0:5,labels=F)
}
if (tsk==tasks[length(tasks)]) {
  mtext(paste("Grade",substr(grd,2,2)),side=4,line=1,xpd=NA)
}
for (ss in 1:nrow(rsdw)) {
  lines(1:3,ssd[ss,paste(c("invartic","invsart","invsartE"),tsk,sep=".")],col="#30303020")
}
boxplot(ssd[,paste("invartic",tsk,sep=".")],at=1,add=T,col="#808000",lwd=1.5,yaxt="n")
boxplot(ssd[,paste("invsart",tsk,sep=".")],at=2,add=T,col="#008080",lwd=1.5,yaxt="n")
boxplot(ssd[,paste("invsartE",tsk,sep=".")],at=3,add=T,col="#800080",lwd=1.5,yaxt="n")
}
}

```



Clearly, the expectation of equal articulation rates between the two task formats does not hold. Serial articulation times per item are on average longer than discrete articulation times for all types of content, though substantial individual variability is evident as some participants achieve higher articulation rates (shorter times) in serial than in discrete tasks. Adjusted times/rates make very little difference, as most individual performance lines are horizontal between raw and adjusted serial times/rates, with the exception of objects.

5.3 Graphs of means for each format

The following graphs display mean articulation time per item for each task and grade.

```

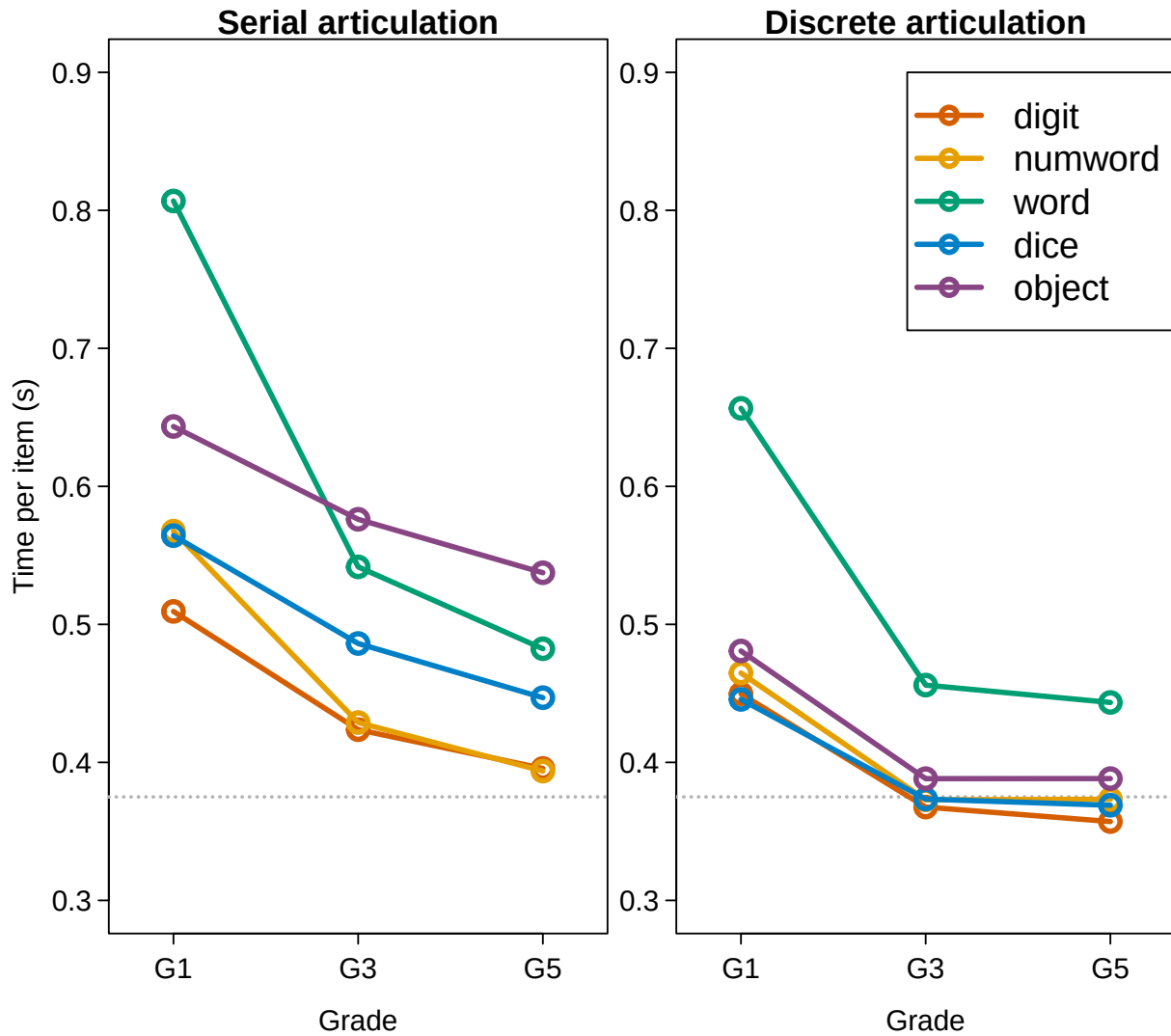
par(mfrow=c(1,2))
par(oma=c(2,2,0,0))
par(mai=c(0.2,0.3,0.2,0.1))
par(mgp=c(2,0.5,0))
par(tcl= -0.3)

serart <- aggregate( as.formula( paste("cbind(", paste(paste("itmarttE",tasks,sep="."),collapse=""), "
plot(NA,NA,type="n",xlim=c(0.75,3.25),ylim=c(0.3,0.9),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("itmarttE",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)
}

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(0.3,0.9),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)
}

legend(1.9,0.9,tasks,col=tcols,pch=21,lty=1,lwd=3,cex=1.25)

```



The horizontal dashed line at $y=0.375$ is added to facilitate comparisons.

5.4 Statistical comparisons between formats (serial vs. discrete)

The following table displays the results of comparing articulation time between the serial and discrete format for each task and grade.

```

artd <- rsdw[,c("sID", "grade", paste("itmarttE", tasks, sep="."), paste("artic", tasks, sep="."))]
artd[,paste("artic", tasks, sep=".")] <- artd[,paste("artic", tasks, sep=".")] / 1000.0
artl1 <- reshape(artd, direction="long",
  varying=list(serial=paste("itmarttE", tasks, sep="."), discrete=paste("artic", tasks, sep=".")),
  v.names=c("serial", "discrete"),
  timevar="task", times=tasks, idvar=c("sID", "grade"))
)
artl <- reshape(artl1, direction="long",
  varying=list(format=c("serial", "discrete")), v.names=c("artic"),
  timevar="format", times=c("serial", "discrete"), idvar=c("sID", "grade", "task"))

artl <- droplevels(subset(artl, (!is.na(artic))))
artl$grade <- relevel(as.factor(artl$grade), ref="G3")

```

```

artl$task <- relevel(as.factor(artl$task),ref="word")
artl$format<- relevel(as.factor(artl$format),ref="discrete")
artl$logartic <- log(artl$artic) # Log-transform to approach a normal distribution
l2 <- lme(logartic~format*task*grade,random=~1|sID/task/format,data=artl)
#anova(l2)
#print(summary(l2),corr=F)
#
## Alternative way of obtaining the contrast matrix via emmeans (confirmed identical result)
#emmeans(l2,pairwise-format/task*grade) -> fpairs # format pairs
#contrast(fpairs[[1]],method="pairwise",adjust="mvt") -> fint
#concoef <- as.list(fint)$linfct
#conname <- apply(expand.grid(levels(summary(fint)$task),levels(summary(fint)$grade),gsub(" ","",levels
#rownames(concoef)<-conname
#
contrast.matrix.serdis <- rbind(
#
#          3dw 3sw          1dw 5dw          1sw 5sw
#          In SER 3dc 3dd 3dn 3do G1 G5 3sc 3sd 3sn 3so SG1 SG5 1dc 1dd 1dn 1do 5dc 5dd 5dn 5
#          1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
"G1 word s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
"G1 dice s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
"G1 digi s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
"G1 numw s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
"G1 obje s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0,

"G3 word s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
"G3 dice s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
"G3 digi s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
"G3 numw s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
"G3 obje s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

"G5 word s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0,
"G5 dice s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0,
"G5 digi s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0,
"G5 numw s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0,
"G5 obje s_d"= c( 0, +1, 0, 0, 0, 0, 0, 0, 0, 0, 0, +1, 0, +1, 0, 0, 0, 0, 0, 0, 0, 0,
)
summary(glht(l2,contrast.matrix.serdis))

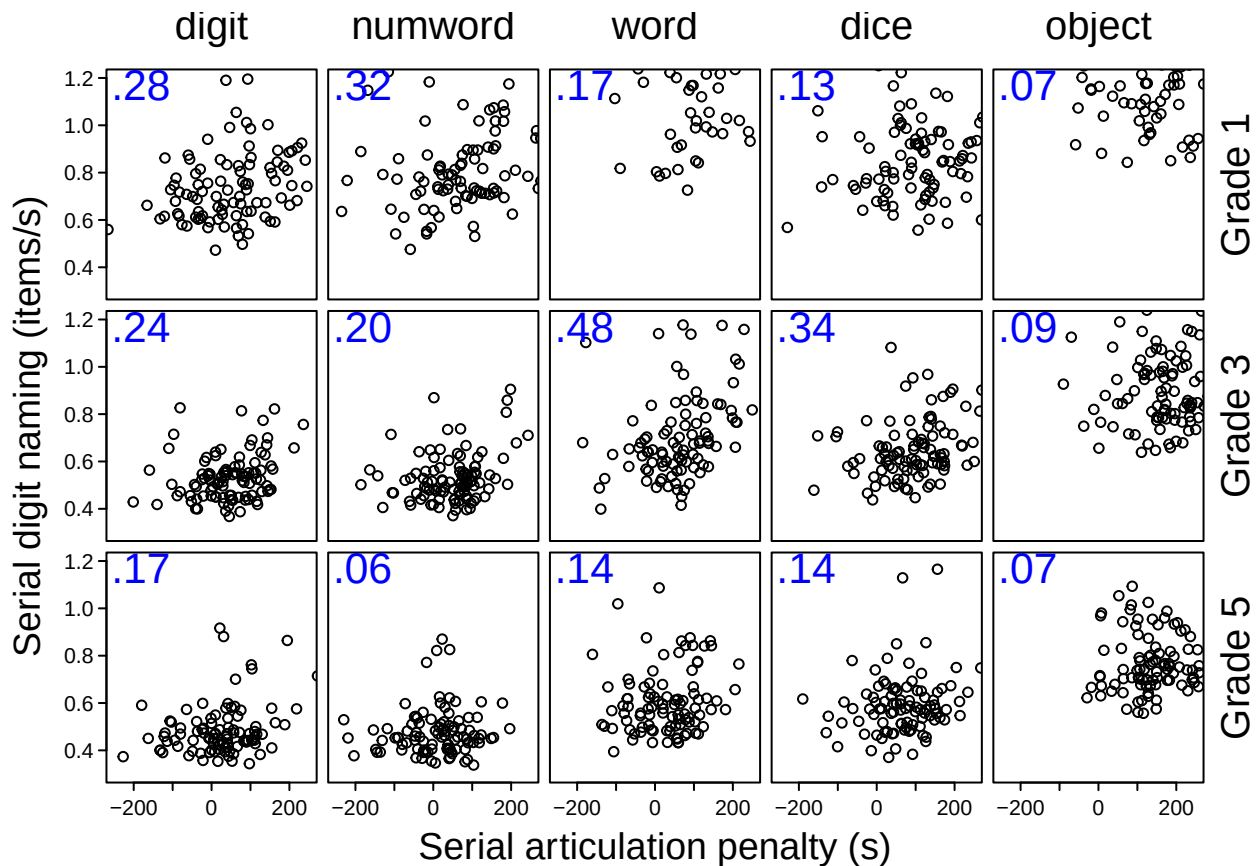
##
## Simultaneous Tests for General Linear Hypotheses
##
## Fit: lme.formula(fixed = logartic ~ format * task * grade, data = artl,
## random = ~1 | sID/task/format)
##
## Linear Hypotheses:
## Estimate Std. Error z value Pr(>|z|)
## G1 word s_d == 0 0.18284 0.01962 9.317 < 1e-10 ***
## G1 dice s_d == 0 0.23040 0.01957 11.774 < 1e-10 ***
## G1 digi s_d == 0 0.12105 0.01962 6.168 1.04e-08 ***
## G1 numw s_d == 0 0.18717 0.01957 9.565 < 1e-10 ***
## G1 obje s_d == 0 0.28724 0.01951 14.721 < 1e-10 ***
## G3 word s_d == 0 0.14875 0.01922 7.739 < 1e-10 ***
## G3 dice s_d == 0 0.24266 0.01933 12.554 < 1e-10 ***
## G3 digi s_d == 0 0.12589 0.01927 6.531 9.78e-10 ***

```

There is a significant serial articulation penalty for every task in all grades except number words in Grade 5. That is, responses take longer to be articulated (longer duration of spoken intervals) in the serial than in the discrete formats, even though it is the exact same responses that are uttered (except for words).

The serial articulation penalty is defined here as the difference between serial and discrete articulation time (per item). As established above, it is positive and statistically significant when the task is cognitively demanding.

```
for (tsk in tasks) {  
  rsdw[,paste("serartpen",tsk,sep=".")] <- rsdw[,paste("itmarttE",tsk,sep=".")]*1000 - rsdw[,paste("articsec",tsk,sep=".")]  
}  
  
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)  
  
for (grd in levels(seradv$grade)) {  
  for (tsk in tasks) {  
    # plot(as.formula(paste("wflut",paste("serartpen",tsk,sep=". "),sep=~")),  
    plot(as.formula(paste(paste("itmadj",tsk,sep=". "),paste("serartpen",tsk,sep=". "),sep=~)),  
         rsdw,subset= grade==grd,  
         #xlim=c(0,1),ylim=c(0,1),  
         xlim=c(-250,250),ylim=c(0.3,1.2),  
         xaxt="n",yaxt="n")  
    if (grd=="G5") { axis(1) }  
    if (tsk==tasks[1]) { axis(2,las=1) }  
    #abline(0,1,lty=3,col="grey50")  
    #text(0,0.95,substr(sprintf("%.02f",cor(rsdw[rsdw$grade==grd,paste("articsec",tsk,sep=". ")],rsdw[rsdw$grade==grd,paste("itmadj",tsk,sep=". ")]),2),0.95))  
    text(-290,1.15,substr(sprintf("%.02f",cor(rsdw[rsdw$grade==grd,paste("itmadj",tsk,sep=". ")],rsdw[rsdw$grade==grd,paste("articsec",tsk,sep=". ")]),2),1.15))  
    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 articulation penalty (s)", 1, outer=T, cex=1.2, line=2)  
mtext("Serial digit naming (items/s)", 2, outer=T, cex=1.2, line=2.2)
```



Most correlations are fairly low and there is no clear pattern here. There is no indication that tasks most subject to articulation penalty exhibit stronger correlations (indeed some of the lowest correlations appear for dice and objects). It is not likely that articulation penalty is associated with serial naming skill.

6 The one-overlap assumption

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).

6.1 Graph of serial total time vs. discrete onset latency

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

serartpau <- aggregate(
  as.formula(
    paste("cbind(",
      paste(paste(
        paste("itmarttE",tasks,sep="."),
        paste("itmpautE",tasks,sep="."),sep="+"

```

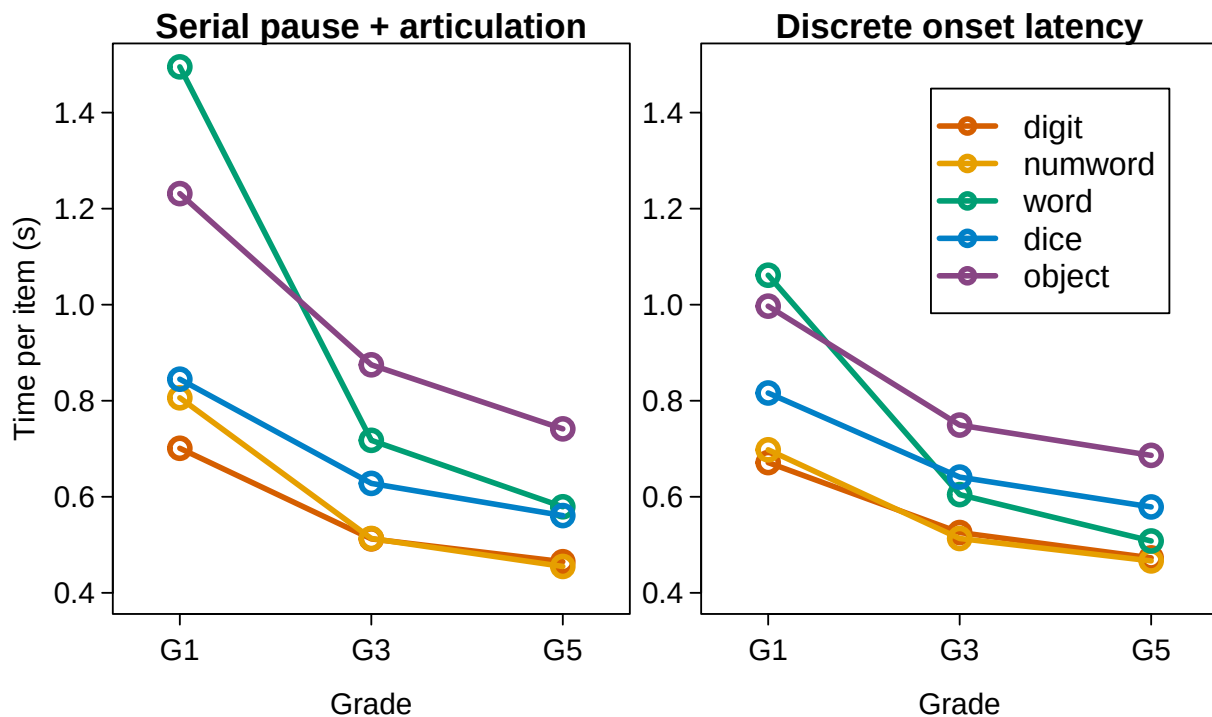
```

    ),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(0.4,1.5),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=0.46,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)
}

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(0.4,1.5),xaxt="n",xlab="Grade",ylab="",las=1,main="Discrete onset latency",
axis(1,at=1:3,labels=paste("G",c(1,3,5),sep=""),cex=1.5)
#abline(h=0.46,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)
}

legend(1.85,1.45,tasks,col=tcols,pch=21,lty=1,lwd=3,cex=1.05)

```



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.

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

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

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("itmpautE",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.

6.3 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=".")] )
}

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 <- "onsRTsec"
v2s <- "serartpau"

for (grd in levels(seradv$grade)) {
  for (tsk in tasks) {

    plot(as.formula(paste(paste(v1s,tsk,sep="."),paste(v2s,tsk,sep="."),sep="~")),
         rsdw,subset= grade==grd,
         xlim=c(0.30,1.75),ylim=c(0.3,1.5),
         xaxt="n",yaxt="n")
    if (grd=="G5") { axis(1) }
  }
}

```

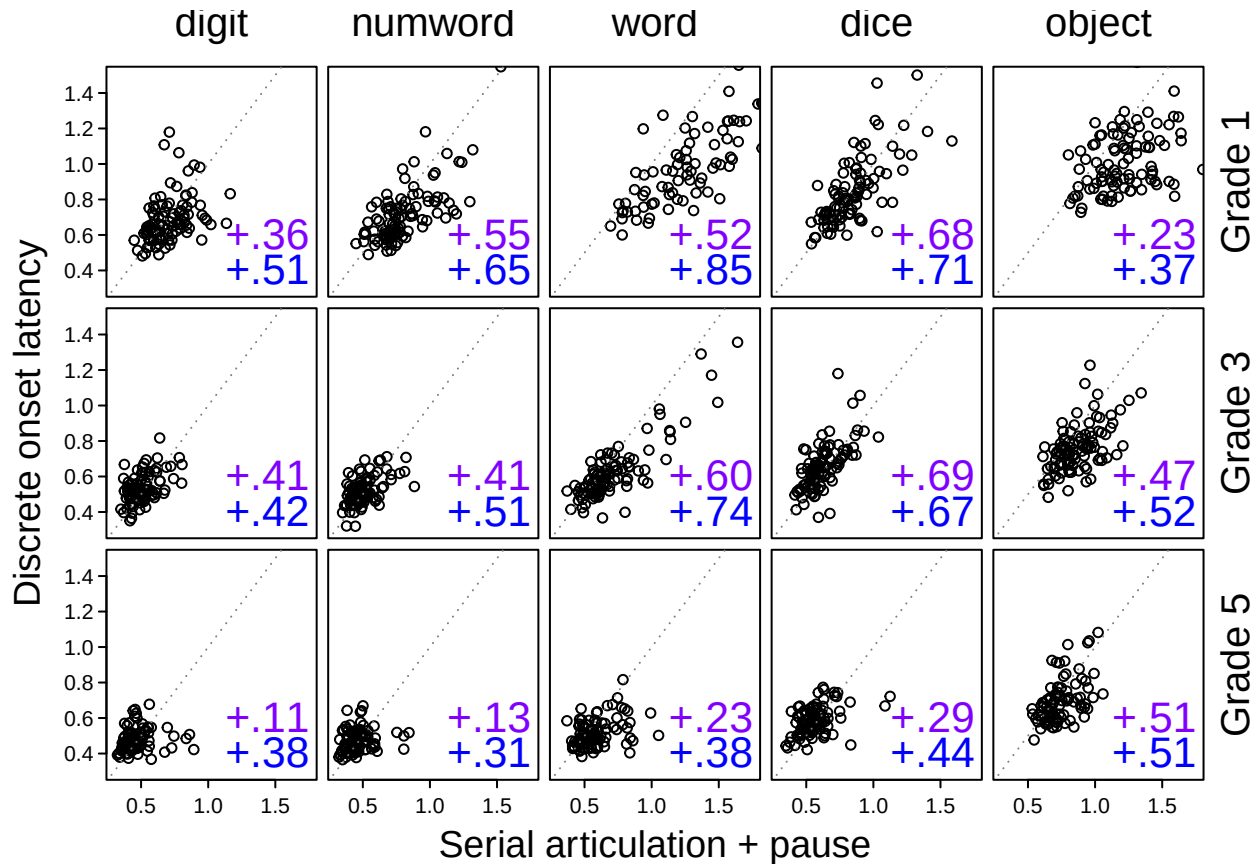
```

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(1.85,0.38,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(1.85,0.58,lmbetastr,cex=2,pos=2,col="#8000FF")

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 articulation + pause", 1, outer=T, cex=1.2, line=2)
mtext("Discrete onset latency", 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 Session information

```
date()
```

```
## [1] "Fri Feb  9 12:34:30 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      multcomp_1.4-24 TH.data_1.1-2    MASS_7.3-60      survival_3.5-5   mvtnorm_1.2-2
```

```
## [7] 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         highr_0.10        zoo_1.8-12
```

```
## [7] colorspace_2.1-0 htmltools_0.5.5   scales_1.2.1      rmarkdown_2.22    grid_4.3.1        munsell_0.5.8
```

```
## [13] evaluate_0.21    fastmap_1.1.1     lifecycle_1.0.3   yaml_2.3.7        compiler_4.3.1    codetools_0.2-19
```

```
## [19] sandwich_3.0-2   rstudioapi_0.14   lattice_0.21-8    digest_0.6.31     R6_2.5.1           mnormt_2.1-1
```

```
## [25] parallel_4.3.1   splines_4.3.1     Matrix_1.5-4.1    tools_4.3.1
```