Experiment 3: Time_on_task

Contents

Introduction

In a preregistered study we manipulated the influence of time on task (Self-paced vs. Timed (3 s) on the Sans Forgetica effect/disfluency effect. We collected 232 participants (116 in each group) on Prolific. This file explains how to read in the data, analyze, and plot the results.

Load in packages

```
#packages you will need
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

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

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(here)
```

```
## here() starts at /Users/jgeller1/Desktop/SF-Testing-New1
```

```
library(afex)
```

```
## Loading required package: lme4
```

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

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

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Registered S3 methods overwritten by 'car':
##   method                          from
##   influence.merMod                lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod         lme4
##   dfbetas.influence.merMod        lme4
```

```
## ************
## Welcome to afex. For support visit: http://afex.singmann.science/
```

```
## - Functions for ANOVAs: aov_car(), aov_ez(), and aov_4()
## - Methods for calculating p-values with mixed(): 'KR', 'S', 'LRT', and 'PB'
## - 'afex_aov' and 'mixed' objects can be passed to emmeans() for follow-up tests
## - NEWS: library('emmeans') now needs to be called explicitly!
## - Get and set global package options with: afex_options()
## - Set orthogonal sum-to-zero contrasts globally: set_sum_contrasts()
## - For example analyses see: browseVignettes("afex")
## ************
```

```
##
## Attaching package: 'afex'
```

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

```
library(Rmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: plyr
```

```
## ------------------------------------------------------------------------------
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## ------------------------------------------------------------------------------
##
## Attaching package: 'plyr'
## The following object is masked from 'package:here':
##
##     here
## The following objects are masked from 'package:dplyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
## The following object is masked from 'package:purrr':
##
##     compact
```

```
library(cowplot)
library(patchwork)
```

```
##
## Attaching package: 'patchwork'
## The following object is masked from 'package:cowplot':
##
##     align_plots
```

```
library(see)
library(ggrepel)
library(report)
```

```
## report is in alpha - help us improve by reporting bugs on github.com/easystats/report/issues
```

```
library(lrd)
```

```
##
## Attaching package: 'lrd'
## The following object is masked from 'package:base':
##
##     kappa
```

```
library(emmeans)
library(MOTE)
library(BayesFactor)
```

```
## Loading required package: coda
## ************
## Welcome to BayesFactor 0.9.12-4.2. If you have questions, please contact Richard Morey (richarddmorey
##
## Type BFManual() to open the manual.
## ************
```

```
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
source("https://gist.githubusercontent.com/benmarwick/2a1bb0133ff568cbe28d/raw/fb53bd97121f7f9ce947837e
```

Read in Data

First we must read in data for each counterbalanced list. There were four and there was no simple way to counterbalance online with PsychoPy and Pavlovia.

CB 1 (Self-Paced)

```
dataset1 <- datasetlow1 %>%
    dplyr::group_by(participant)%>%
    dplyr::filter(mouse_5.clicked_name=="polygon_2") %>% dplyr::select(textbox.text, cue1, targ1, font)
    mutate(textbox.text=tolower(textbox.text), acc=ifelse(textbox.text==cue1, 1, 0)) %>%
  mutate(cond="self-paced")
```

Extract cued responses CB1

```
## Adding missing grouping variables: `participant`
```

CB 2 (Self-paced)

```
dataset2 <- datasetlow2 %>%
    dplyr::group_by(participant)%>%
    dplyr::filter(mouse_5.clicked_name=="polygon_2") %>% dplyr::select(textbox.text, cue1, targ1, font)
    mutate(textbox.text=tolower(textbox.text), acc=ifelse(textbox.text==cue1, 1, 0)) %>%
  mutate(cond="self-paced")
```

Extract cued responses (CB2)

```
## Adding missing grouping variables: `participant`
```

CB 1 (Timed)

```
dataset3 <- datasetlow3 %>%
    dplyr::group_by(participant)%>%
    dplyr::filter(mouse_5.clicked_name=="polygon_2") %>% dplyr::select(textbox.text, cue1, targ1, font)
    mutate(textbox.text=tolower(textbox.text), acc=ifelse(textbox.text==cue1, 1, 0)) %>%
  mutate(cond="timed")
```

Extract cued responses

```
## Adding missing grouping variables: `participant`
```

CB 2 (Timed)

```
dataset4 <- datasetlow4 %>%
    dplyr::group_by(participant)%>%
    dplyr::filter(mouse_5.clicked_name=="polygon_2") %>% dplyr::select(textbox.text, cue1, targ1, font)
    mutate(textbox.text=tolower(textbox.text), acc=ifelse(textbox.text==cue1, 1, 0)) %>%
  mutate(cond="timed")
```

Extract cued responses

```
## Adding missing grouping variables: `participant`
```

Combine Data

Cued Responses

```
all<-rbind(dataset1, dataset2, dataset3, dataset4)

#write.csv(all, file="all.csv")
```

JOls

Extract JOLs and combine them

```
dataset1_jol <- datasetlow1 %>%
    dplyr::group_by(participant)%>%
     dplyr::select(participant, atypic_slider.response, normal_slider.response) %>%
  mutate(cond="self-paced") %>%
  na.omit(.) %>%
  pivot_longer(atypic_slider.response:normal_slider.response, names_to = "Typeface", values_to = "jols")

dataset2_jol <- datasetlow2 %>%
    dplyr::group_by(participant)%>%
     dplyr::select(participant, atypic_slider.response, normal_slider.response) %>%
  mutate(cond="self-paced") %>%
  na.omit(.) %>%  pivot_longer(atypic_slider.response:normal_slider.response, names_to = "Typeface", val


dataset3_jol <- datasetlow3 %>%
    dplyr::group_by(participant)%>%
     dplyr::select(participant, atypic_slider.response, normal_slider.response) %>%
  mutate(cond="timed") %>%
  na.omit(.) %>%
pivot_longer(atypic_slider.response:normal_slider.response, names_to = "Typeface", values_to = "jols")

dataset4_jol <- datasetlow4 %>%
    dplyr::group_by(participant)%>%
     dplyr::select(participant, atypic_slider.response, normal_slider.response) %>%
  mutate(cond="timed") %>%
  na.omit(.) %>%
  pivot_longer(atypic_slider.response:normal_slider.response, names_to = "Typeface", values_to = "jols")

jol_all <- rbind(dataset1_jol, dataset2_jol, dataset3_jol, dataset4_jol)
```

Analysis

Cued Recall

Read in Scored file

Takes data from Shiny LRD package.

Cued Response Scoring

The LRD package scores cued response data. The data is loaded into a shiny application and then brought back in to R. Below you can get proprotion correct by participant or the the trial-level Scored data,

```
#devtools::install_github("npm27/lrd") # load the package
#library(lrd)

all$textbox.text[is.na(all$textbox.text)] <- "" # does not work if NAs exists

all_c<-as.data.frame(all) # needs to be data frame

all_c$trial_id<-rep(1:5568) # needs to have unique rows for some reason
# run lrd
scored_cued = prop_correct_cued(all_c,
                                responses = "textbox.text",
                                id = "participant",
                                id.trial = "trial_id",
                                key = "targ1",
                                key.trial =  "trial_id",
                                cutoff = 3,
                                group.by = c("font", "cond"))

recall_time<-scored_cued$DF_Participant # Prop by participant

recall_time<-scored_cued$Scored # trial-level (0,1)
```

Difference

```
recall_all1_diff <- recall_time %>%
  dplyr::group_by(id, font, cond)%>%
  dplyr::summarise(acc=mean(Scored)) %>%
  tidyr::pivot_wider(names_from="font", values_from="acc") %>%
  mutate(Difference=SF-flu, cond=ifelse(cond=="self-paced","Self-paced", "Timed(3s)"))
```

```
## `summarise()` has grouped output by 'id', 'font'. You can override using the `.groups` argument.
```

```
recall_all_mean <- recall_all1_diff %>%
  dplyr::group_by(cond)%>%
  dplyr::summarise(mean=mean(Difference))

write.csv(recall_all1, file="wide_recall_timed.csv")


#write.csv(recall_all1, file="long_recall_timed.csv")
```

ANOVA

```
#ANOVA


a1 <- aov_ez("id", "acc", recall_all1,
             within=c("font"), between=c("cond")) # mixed
```

```
## Converting to factor: cond
```

```
## Contrasts set to contr.sum for the following variables: cond
```

```
a1
```

```
## Anova Table (Type 3 tests)
##
## Response: acc
##      Effect     df  MSE         F   ges p.value
## 1      cond 1, 230 0.09      0.37  .001    .544
## 2      font 1, 230 0.02 15.04 ***  .013   <.001
## 3 cond:font 1, 230 0.02      1.13 <.001    .289
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
# Main Effects
```

Main Effects

Get means for each main effect

```
Within_font <- emmeans(a1, ~ font)
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
Within_font
```

```
##  font emmean     SE  df lower.CL upper.CL
##  flu   0.323 0.0152 337    0.293    0.353
##  SF    0.375 0.0152 337    0.345    0.405
##
## Results are averaged over the levels of: cond
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

```
Within_cond <- emmeans(a1, ~ cond)
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
Within_cond
```

```
##  cond       emmean     SE  df lower.CL upper.CL
##  self-paced  0.341 0.0192 230    0.303    0.379
##  timed       0.357 0.0192 230    0.320    0.395
##
## Results are averaged over the levels of: font
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

Interaction

```
Within_Fitted_Interaction <- emmeans(a1, ~ font|cond)
Within_Fitted_Interaction
```

```
## cond = self-paced:
##  font emmean     SE  df lower.CL upper.CL
##  flu   0.322 0.0215 337    0.280    0.364
##  SF    0.360 0.0215 337    0.318    0.402
##
## cond = timed:
##  font emmean     SE  df lower.CL upper.CL
##  flu   0.324 0.0215 337    0.282    0.366
##  SF    0.391 0.0215 337    0.349    0.433
##
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

Planned Comparisons

```
pairs(Within_Fitted_Interaction) ## pairwise comparison with no correction
```

```
## cond = self-paced:
##  contrast estimate     SE  df t.ratio p.value
##  flu - SF  -0.0381 0.0191 230 -1.991  0.0477
##
## cond = timed:
##  contrast estimate     SE  df t.ratio p.value
##  flu - SF  -0.0668 0.0191 230 -3.493  0.0006
```

Effect sizes

```
recall_sp <- recall_all1 %>%
  dplyr::group_by(id, font, cond) %>%
  dplyr::summarise(mean_acc=mean(acc))%>%
  tidyr::pivot_wider(names_from = "font", values_from = "mean_acc")%>%
dplyr::filter(cond=="self-paced")%>%
  ungroup() %>%
  summarise(mean1=mean(flu), sd1=sd(flu), mean2=mean(SF), sd2=sd(SF))
```

```
## `summarise()` has grouped output by 'id', 'font'. You can override using the `.groups` argument.
```

```
sp_recall=d.dep.t.avg(m1 = recall_sp$mean1, m2 = recall_sp$mean2, sd1 = recall_sp$sd1,
                sd2 = recall_sp$sd2, n = 116, a = .05)

recall_timed <- recall_all1 %>%
  dplyr::group_by(id, font, cond) %>%
  dplyr::summarise(mean_acc=mean(acc))%>%
  tidyr::pivot_wider(names_from = "font", values_from = "mean_acc")%>%
dplyr::filter(cond=="timed")%>%
  ungroup() %>%
  summarise(mean1=mean(flu), sd1=sd(flu), mean2=mean(SF), sd2=sd(SF))
```

```
## `summarise()` has grouped output by 'id', 'font'. You can override using the `.groups` argument.
```

x	x	x	x	x	x	x	x
-0.1500701	-0.3327541	0.0332587	0.3218391	0.2593415	0.0240793	0.2741427	0.3695355

x	x	x	x	x	x	x
0.3599138	0.2480842	0.023034	0.3142878	0.4055398	116	115

x
d_{av} = -0.15, 95\% CI [-0.33, 0.03]

x	x	x	x	x	x	x
-0.3238292	-0.5098777	-0.1364414	0.3239943	0.1974388	0.0183317	0.2876826

x	x	x	x	x	x	x	x
0.3603059	0.3908046	0.2151883	0.0199797	0.3512286	0.4303806	116	115

x
d_{av} = -0.32, 95\% CI [-0.51, -0.14]

```
time_recall=d.dep.t.avg(m1 = recall_timed$mean1, m2 = recall_timed$mean2, sd1 = recall_timed$sd1,
                sd2 = recall_timed$sd2, n = 116, a = .05)

kable(sp_recall)
```

```
kable(time_recall)
```

Bayesian Analyis

Cued Recall

Run the 2 x 2 Bayesian Analyis

```
recall_all1$new_id<-rep(1:232, each=2)
recall_all1$new_id<-as.factor(recall_all1$new_id)
recall_all1$cond<-as.factor(recall_all1$cond)
recall_all1$font<-as.factor(recall_all1$font)

bfcue = anovaBF(acc ~ cond*font + new_id, recall_all1,
          whichRandom=c("new_id"))
```

JOLS

ANOVA A 2 x 2 Mixed ANOVA was run.

```
#ANOVA
a1 <- aov_ez("participant", "jols", jol_all,
            within=c("Typeface"), between=c("cond")) # mixed
```

```
## Converting to factor: cond
```

```
## Contrasts set to contr.sum for the following variables: cond
```

```
summary(a1)
```

```
##
## Univariate Type III Repeated-Measures ANOVA Assuming Sphericity
##
##                Sum Sq num Df Error SS den Df  F value    Pr(>F)
## (Intercept)   1887931      1   165126    230 2629.662 < 2.2e-16 ***
## cond            12514      1   165126    230   17.430 4.230e-05 ***
```

```
## Typeface        10871      1    51231    230   48.806 3.016e-11 ***
## cond:Typeface    6053      1    51231    230   27.173 4.138e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
a1
```

```
## Anova Table (Type 3 tests)
##
## Response: jols
##          Effect     df    MSE         F  ges p.value
## 1          cond 1, 230 717.94 17.43 *** .055   <.001
## 2      Typeface 1, 230 222.74 48.81 *** .048   <.001
## 3 cond:Typeface 1, 230 222.74 27.17 *** .027   <.001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

Main Effects

```
Within_font <- emmeans(a1, ~ Typeface)
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
Within_font
```

```
##  Typeface               emmean   SE  df lower.CL upper.CL
##  atypic_slider.response   58.9 1.42 360     56.1     61.7
##  normal_slider.response   68.6 1.42 360     65.8     71.4
##
## Results are averaged over the levels of: cond
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

```
Within_cond <- emmeans(a1, ~ cond)
```

```
## NOTE: Results may be misleading due to involvement in interactions
```

```
Within_cond
```

```
##  cond       emmean   SE  df lower.CL upper.CL
##  self-paced   58.6 1.76 230     55.1     62.1
##  timed        69.0 1.76 230     65.5     72.4
##
## Results are averaged over the levels of: Typeface
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

Testing Interaction

```
Within_Fitted_Interaction <- emmeans(a1, ~ Typeface|cond)
Within_Fitted_Interaction
```

```
## cond = self-paced:
##  Typeface               emmean   SE  df lower.CL upper.CL
##  atypic_slider.response   50.1 2.01 360     46.2     54.1
##  normal_slider.response   67.0 2.01 360     63.1     71.0
##
## cond = timed:
```

```
##  Typeface               emmean   SE  df lower.CL upper.CL
##  atypic_slider.response   67.8 2.01 360     63.8     71.7
##  normal_slider.response   70.2 2.01 360     66.2     74.2
##
## Warning: EMMs are biased unless design is perfectly balanced
## Confidence level used: 0.95
```

Effect sizes

```
recall_spjol <- jol_all %>%
  dplyr::group_by(participant, Typeface, cond) %>%
  dplyr::summarise(mean_jol=mean(jols))%>%
  tidyr::pivot_wider(names_from = "Typeface", values_from = "mean_jol")%>%
dplyr::filter(cond=="self-paced")%>%
  ungroup() %>%
  summarise(mean1=mean(normal_slider.response ), sd1=sd(normal_slider.response), mean2=mean(atypic_slid
```

```
## `summarise()` has grouped output by 'participant', 'Typeface'. You can override using the `.groups` a
```

```
sp_jol=d.dep.t.avg(m1 = recall_spjol$mean1, m2 = recall_spjol$mean2, sd1 = recall_spjol$sd1,
                sd2 = recall_spjol$sd2, n = 116, a = .05)

recall_timejol <- jol_all %>%
  dplyr::group_by(participant, Typeface, cond) %>%
  dplyr::summarise(mean_jol=mean(jols))%>%
  tidyr::pivot_wider(names_from = "Typeface", values_from = "mean_jol")%>%
dplyr::filter(cond=="timed")%>%
  ungroup() %>%
  summarise(mean1=mean(normal_slider.response ), sd1=sd(normal_slider.response), mean2=mean(atypic_slid
```

```
## `summarise()` has grouped output by 'participant', 'Typeface'. You can override using the `.groups` a
```

```
t_jol=d.dep.t.avg(m1 = recall_timejol$mean1, m2 = recall_timejol$mean2, sd1 = recall_timejol$sd1,
                sd2 = recall_timejol$sd2, n = 116, a = .05)
sp_jol
```

```
## $d
## [1] 1.218282
##
## $dlow
## [1] 0.9761973
##
## $dhigh
## [1] 1.457231
##
## $M1
## [1] 67.0462
##
## $sd1
## [1] 26.78193
##
## $se1
## [1] 2.48664
##
## $M1low
```

```
## [1] 62.12065
##
## $M1high
## [1] 71.97176
##
## $M2
## [1] 50.14192
##
## $sd2
## [1] 0.9690968
##
## $se2
## [1] 0.08997838
##
## $M2low
## [1] 49.96369
##
## $M2high
## [1] 50.32015
##
## $n
## [1] 116
##
## $df
## [1] 115
##
## $estimate
## [1] "$d_{av}$ = 1.22, 95\\% CI [0.98, 1.46]"
```

```
t_jol
```

```
## $d
## [1] 0.1019037
##
## $dlow
## [1] -0.08076949
##
## $dhigh
## [1] 0.2841364
##
## $M1
## [1] 70.20921
##
## $sd1
## [1] 23.89936
##
## $se1
## [1] 2.219
##
## $M1low
## [1] 65.8138
##
## $M1high
## [1] 74.60462
##
```

```
## $M2
## [1] 67.75181
##
## $sd2
## [1] 24.3305
##
## $se2
## [1] 2.25903
##
## $M2low
## [1] 63.27711
##
## $M2high
## [1] 72.22652
##
## $n
## [1] 116
##
## $df
## [1] 115
##
## $estimate
## [1] "$d_{av}$ = 0.10, 95\\% CI [-0.08, 0.28]"
```

Bayesian Analysis

```
jol_all$new_id1<-rep(1:232, each=2)
jol_all$new_id1<-as.factor(jol_all$new_id1)
jol_all$cond<-as.factor(jol_all$cond)
jol_all$Typeface<-as.factor(jol_all$Typeface)
bftimed = anovaBF(jols ~ cond*Typeface, jol_all,
          whichRandom="new_id1")
```

Study Times

Reviewers wanted to see study time data

Dependent t test

```
# RTs for self-paced group

dataset1_rt <- datasetlow1 %>%
    dplyr::group_by(participant)%>%
     dplyr::select(participant, mouse_4.time, font) %>%
  dplyr::mutate(cond="self-paced")%>%
  dplyr::ungroup() %>%
  dplyr::select(participant, cond, font, mouse_4.time)


dataset2_rt <-  datasetlow2 %>%
    dplyr::group_by(participant)%>%
     dplyr::select(participant, mouse_4.time, font) %>%
```

```
  dplyr::mutate(new_id=ifelse(is.na(participant), turkid, participant)) %>%
  dplyr::mutate(new_id1=ifelse(is.na(new_id), participant, new_id))%>%
  dplyr::mutate(cond="self_paced")%>%
  dplyr::ungroup() %>%
  dplyr::select(participant, cond, font, mouse_4.time)

#as.data.frame(dplyr::count(dataset3_rt, new_id1))

dataset3_all<-rbind(dataset1_rt, dataset2_rt)

dataset3_all <- na.omit(dataset3_all)

rt_all_wide<- dataset3_all %>%
  dplyr::group_by(participant, font, cond) %>%
  dplyr::mutate(rt=mouse_4.time*1000) %>%
dplyr::mutate(sdabove = mean(rt, na.rm=TRUE) +  2.5*sd(rt, na.rm=TRUE)) %>%
    dplyr::filter(rt > 150 || rt > sdabove) %>%
  dplyr::summarise(mean_rt= mean(log(rt), na.rm=TRUE)) %>%
   mutate(font=ifelse(font=="flu", "Arial", "Sans Forgetica")) %>%
   tidyr::pivot_wider(names_from = "font", values_from = "mean_rt")
```

```
## `summarise()` has grouped output by 'participant', 'font'. You can override using the `.groups` argu
```

```
t.test(rt_all_wide$`Sans Forgetica`, rt_all_wide$Arial, paired=TRUE)
```

```
##
##  Paired t-test
##
## data:  rt_all_wide$`Sans Forgetica` and rt_all_wide$Arial
## t = 4.1096, df = 115, p-value = 7.46e-05
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.03286052 0.09401291
## sample estimates:
## mean of the differences
##              0.06343671
```

Effect Size

```
rt_all1<- dataset3_all%>%
  dplyr::group_by(participant, font, cond) %>%
  dplyr::mutate(rt=mouse_4.time*1000) %>%
dplyr::mutate(sdabove = mean(rt, na.rm=TRUE) +  2.5*sd(rt, na.rm=TRUE)) %>%
    dplyr::filter(rt > 150 || rt > sdabove) %>%
  dplyr::summarise(mean_rt= mean(log(rt), na.rm=TRUE)) %>%
   mutate(font=ifelse(font=="flu", "Arial", "Sans Forgetica")) %>%
   tidyr::pivot_wider(names_from = "font", values_from = "mean_rt")%>%
  ungroup() %>%
  dplyr::summarise(mean1=mean(Arial), mean2=mean(`Sans Forgetica`), sd1=sd(Arial), sd2=sd(`Sans Forgeti
```

```
## `summarise()` has grouped output by 'participant', 'font'. You can override using the `.groups` argu
```

```
rt_h=d.dep.t.avg(m1 = mean(rt_all1$mean1), m2 = mean(rt_all1$mean2), sd1 = mean(rt_all1$sd1),
                 sd2 = mean(rt_all1$sd2), n = 116, a = .05)
```

```
rt_h
```

```
## $d
## [1] -0.1054439
##
## $dlow
## [1] -0.2877027
##
## $dhigh
## [1] 0.0772703
##
## $M1
## [1] 7.253773
##
## $sd1
## [1] 0.6067195
##
## $se1
## [1] 0.0563325
##
## $M1low
## [1] 7.14219
##
## $M1high
## [1] 7.365357
##
## $M2
## [1] 7.31721
##
## $sd2
## [1] 0.5965115
##
## $se2
## [1] 0.0553847
##
## $M2low
## [1] 7.207504
##
## $M2high
## [1] 7.426917
##
## $n
## [1] 116
##
## $df
## [1] 115
##
## $estimate
## [1] "$d_{av}$ = -0.11, 95\\% CI [-0.29, 0.08]"
```

Plot

Cued recall

```
bold <- element_text(face = "bold", color = "black", size = 14)

recall_all1 <- recall_all1 %>%
  mutate(Typeface=ifelse(font=="SF", "Sans Forgetica", ifelse(font=="flu", "Arial", "Difference"))) %>%
  mutate(timed=ifelse(cond=="self-paced", "Self-paced", "Timed"))

#means = recall_all1 %>%
 # dplyr::group_by(timed, Typeface)%>%
 # dplyr::summarise(mean=mean(acc))


sfgen_wsci=summarySEwithin(data = recall_all1, measurevar = "acc",
                       withinvars = "Typeface", betweenvars = "timed", idvar = "id")
```

```
## Automatically converting the following non-factors to factors: timed, Typeface
```

```
fig3a <- ggplot(recall_all1,aes(x=Typeface,y=acc,fill=Typeface))+
  facet_grid(~timed) +
  #geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .4,adjust=4)+
  geom_point(position=position_jitter(width = .15),size = 1, alpha = 0.2) +
  geom_boxplot(aes(x = Typeface, y = acc ),outlier.shape = NA,
               alpha = 0.3, width = .1, colour = "BLACK") +
    #stat_summary(fun="mean", geom="point", colour="darkred", size=3)+
  geom_line(data=sfgen_wsci,aes(y=acc, group=1), size=1)+
  geom_pointrange(data=sfgen_wsci, aes(y=acc, ymin=acc, ymax=acc), size=.8, color="darkred")+
  scale_colour_brewer(palette = "Dark2")+
  scale_fill_brewer(palette = "Dark2") +
  labs(y = "Porportion Correct on Final Test", x = "Typeface") +
  theme(legend.position = "none") +
   geom_label_repel(data=sfgen_wsci, aes(y=acc, label=round(acc, 2)),seed = 42, box.padding = 0.8) +
   theme_cowplot(font_size=14)+
  theme(legend.position = "none") +
  theme(axis.title = bold)

# plot difference plots
fig3a_diff <- ggplot(recall_all1_diff,aes(x=cond,y=Difference, fill=cond)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .4,adjust=4)+
  geom_point(position=position_jitter(width = .18),size = 1, alpha = 0.2) +
  geom_boxplot(aes(x = cond, y = Difference),outlier.shape = NA,
               alpha = 0.3, width = .1, colour = "BLACK") +
  stat_summary(fun.data="mean_cl_boot", colour="darkred", size=.8)+
  #geom_line(data=sfarial_wsci,aes(y=mean_acc, group=1), size=1)+
  #geom_pointrange(data=sfarial_wsci, aes(y=mean_acc, ymin=mean_acc-ci, ymax=mean_acc+ci), size=.5, col
  scale_colour_brewer(palette = "Accent")+
  scale_fill_brewer(palette = "Accent") +
  labs(y = "Test Difference (Sans Forgetica - Arial", x = "Time on Task")+
     theme_cowplot(font_size=14)+
  theme(legend.position = "none") +
  theme(axis.title =bold) +
  geom_hline(yintercept = 0, linetype="dotted") +
   geom_label_repel(data=recall_all_mean, aes(y=mean, label=round(mean, 2)), seed=42, box.padding=0.8)
```

```
fig3a
```

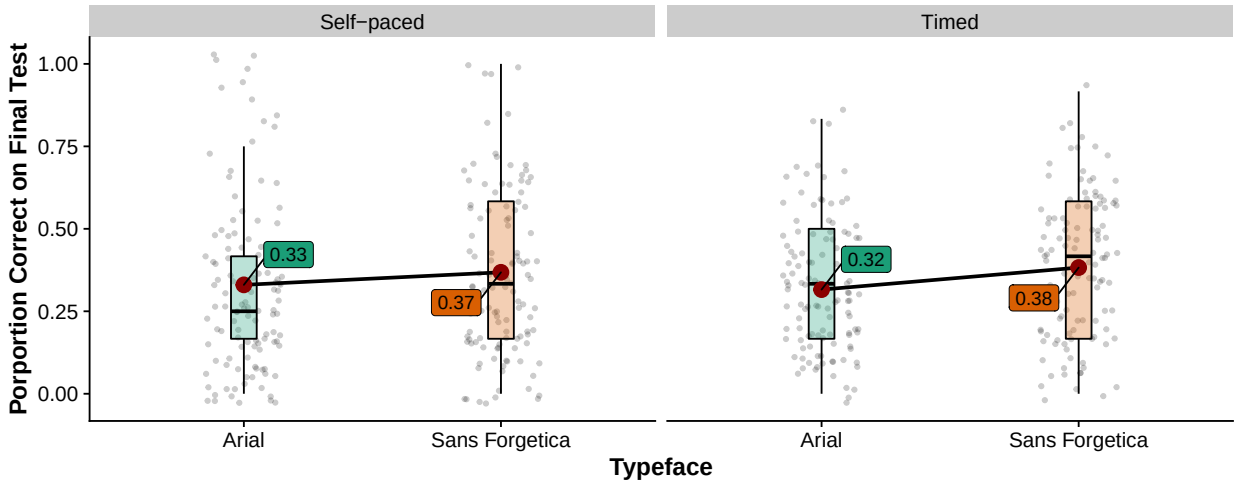

Figure 1: Raincloud plots (Allen et al., 2019) depicting raw data (dots), box plots, and half violin kernel desntiy plots, with mean (red dot) and within-participant 95 CIs. Cued recall accuracy as a function of Time on task for Experiment 3.

```
fig3a_diff
```

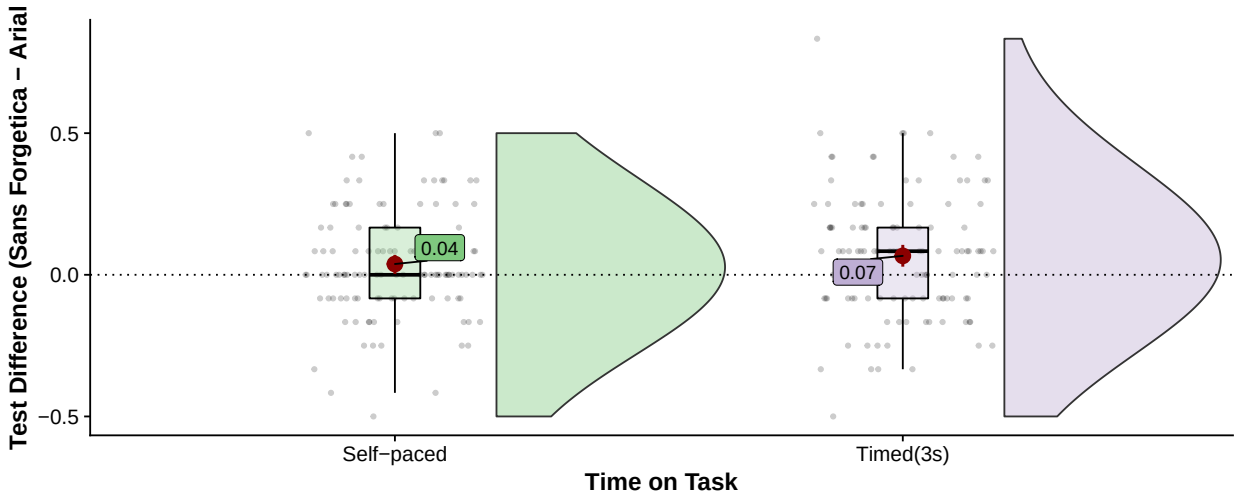

Figure 2: Raincloud plots (Allen et al., 2019) depicting raw data (dots), box plots, and half violin kernel desntiy plots, with mean (red dot) and within-participant 95 CIs. Cued recall accuracy as a function of Time on task for Experiment 3.

JOLS

```
jol_rename <- jol_all %>%
  mutate(Typeface=ifelse(Typeface=="atypic_slider.response", "Sans Forgetica", "Arial")) %>%
  mutate(timed=ifelse(cond=="self-paced", "Self-paced", "Timed"))
```

```
jol_diff <- jol_rename %>%
  pivot_wider(names_from="Typeface", values_from = "jols")%>%
  dplyr::mutate(Difference=`Sans Forgetica`- Arial) %>%
  dplyr::mutate(cond=ifelse(cond=="self-paced", "Self-paced", "Timed(3s)"))

jol_diff_mean <- jol_diff %>%
  dplyr::group_by(cond) %>%
  dplyr::summarise(mean=mean(Difference))

means = jol_rename %>%
  dplyr::group_by(timed, Typeface)%>%
  dplyr::summarise(mean=mean(jols))%>%
  dplyr::mutate(timed=as.factor(timed), Typeface=as.factor(Typeface))
```

```
## `summarise()` has grouped output by 'timed'. You can override using the `.groups` argument.
```

```
# get withinsubject CIs
sfgenjol_wsci=summarySEwithin(data = jol_rename, measurevar = "jols",
                      withinvars = "Typeface", betweenvars = "timed", idvar = "participant")
```

```
## Automatically converting the following non-factors to factors: timed, Typeface
```

```
fig3b <- ggplot(jol_rename,aes(x=Typeface,y=round(jols,2),fill=Typeface))+
  facet_grid(~timed) +
  #geom_violinhalf(position = position_nudge(x = .2, y = 0), alpha = .5,adjust=4)+
  #geom_violinhalf(fill_dots = "black") +
  geom_point(position=position_jitter(width = .15),size = 1, alpha = 0.2) +
  geom_boxplot(aes(x = Typeface, y = jols ),outlier.shape = NA,
               alpha = 0.3, width = .1, colour = "BLACK") +
   #stat_summary(fun="mean", geom="point", colour="darkred", size=3)+
  geom_line(data=sfgenjol_wsci,aes(y=jols, group=1), size=1)+
  geom_pointrange(data=sfgenjol_wsci, aes(y=jols, ymin=jols, ymax=jols), size=.8, color="darkred")+
  theme_cowplot() +
  scale_colour_brewer(palette = "Dark2")+
  scale_fill_brewer(palette = "Dark2") +
  labs(y = "Judgements of Learning", x = "Typeface") +
  theme(legend.position = "none") +
   geom_label_repel(data=sfgenjol_wsci, aes(y=jols, label=round(jols, 2)),seed = 42, box.padding = 0.8)
  theme_cowplot(font_size=14)+
  theme(legend.position = "none") +
  theme(axis.title = bold)



fig3b_diff <- ggplot(jol_diff,aes(x=cond,y=Difference,fill=cond)) +
  geom_flat_violin(position = position_nudge(x = .2, y = 0), alpha = .4,adjust=4)+
  geom_point(position=position_jitter(width = .15),size = 1, alpha = 0.2) +
  geom_boxplot(aes(x = cond, y = Difference),outlier.shape = NA,
               alpha = 0.3, width = .1, colour = "BLACK") +
  stat_summary(fun.data="mean_cl_boot", colour="darkred", size=.8)+
   #stat_summary(fun="mean", geom="point", colour="darkred", size=3)+
 # geom_line(data=sfgenjol_wsci,aes(y=jols, group=1), size=1)+
 # geom_pointrange(data=sfgenjol_wsci, aes(y=jols, ymin=jols-ci, ymax=jols+ci), size=.3, color="red")+
  scale_colour_brewer(palette = "Accent")+
  scale_fill_brewer(palette = "Accent") +
```

```
  labs(y = "JOL Difference (Sans Forgetica - Arial)", x = "Time on Task") +  theme(legend.position = "no
  geom_label_repel(data=jol_diff_mean, aes(y=mean , label=round(mean, 2)), seed = 42, box.padding = 0.8)
   theme_cowplot(font_size=14)+
  theme(legend.position = "none") +
  geom_hline(yintercept = 0, linetype="dotted") +
  theme(axis.title = bold)


fig3b
```

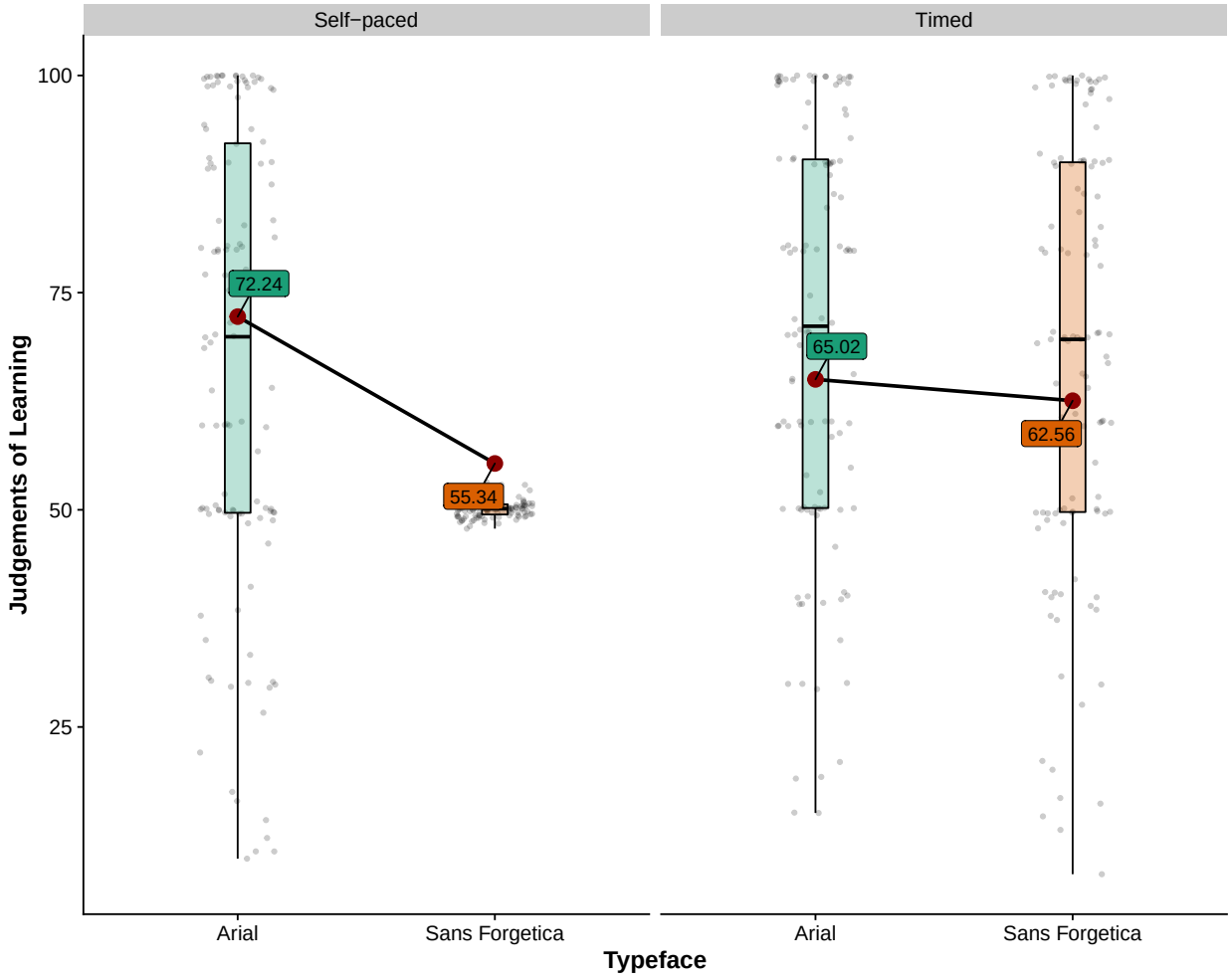

Figure 3: Raincloud plots (Allen et al., 2019) depicting raw data (dots), box plots, and half violin kernel desntiy plots, with mean (red dot) and within-participant 95 CIs. Cued recall accuracy as a function of Time on task for Experiment 3.

```
fig3b_diff
```

Combine Plots

```
fig3 <- plot_grid(
  fig3a, fig3b,
  labels = "AUTO", ncol= 1, nrow = 2
```

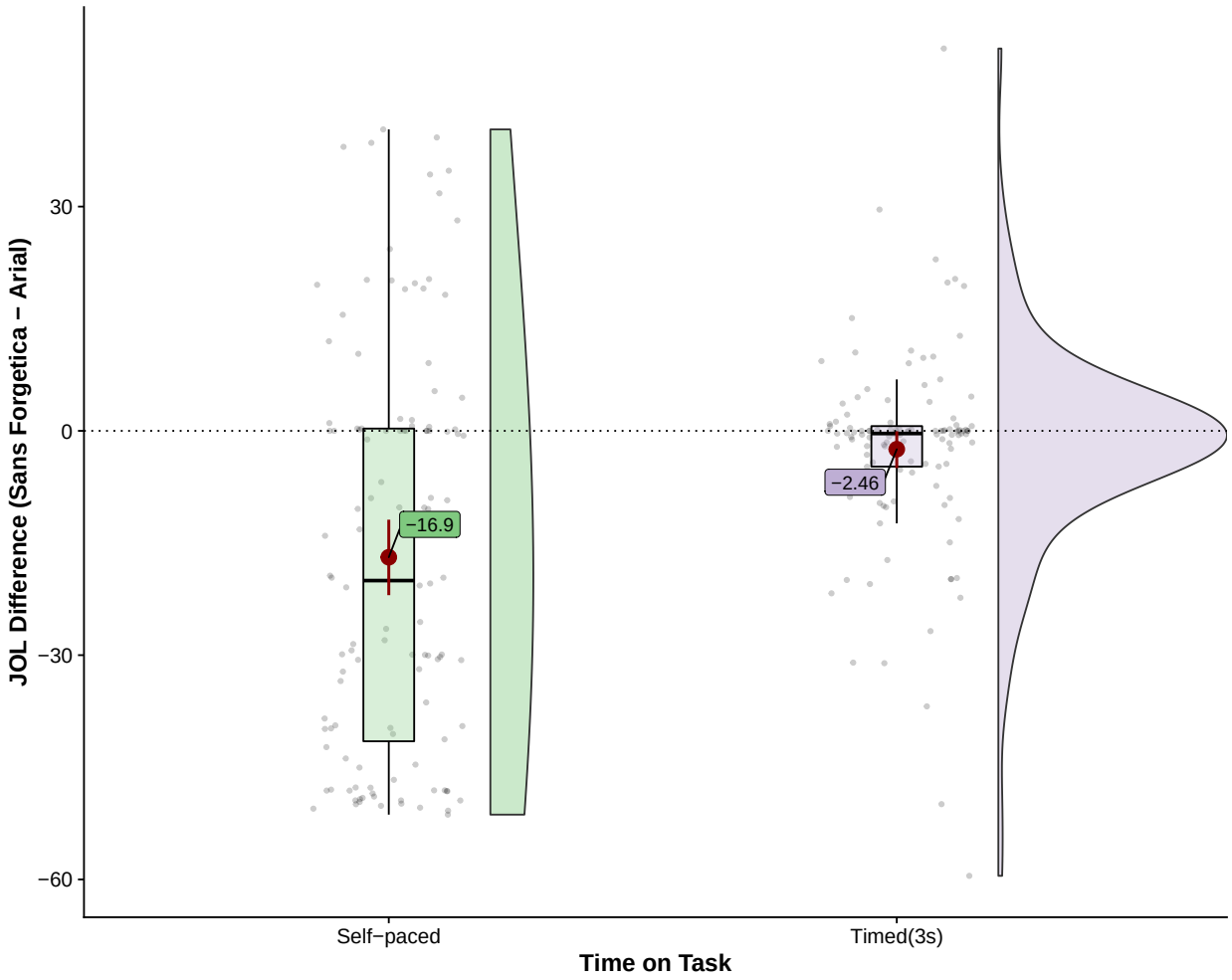

Figure 4: Raincloud plots (Allen et al., 2019) depicting raw data (dots), box plots, and half violin kernel desntiy plots, with mean (red dot) and within-participant 95 CIs. Cued recall accuracy as a function of Time on task for Experiment 3.

```
)

ggsave("fig3experiment3.png", width=10, height=12, dpi=500)

fig3_diff <- plot_grid(fig3a_diff, fig3b_diff, ncol=, nrow=2)

ggsave("fig3_diff.png", width=10, height=12, dpi=500)


fig3_diff <- plot_grid(fig3a, fig3a_diff, fig3b, fig3b_diff, ncol=2, nrow=2)

ggsave("fig3_diff_all.png", width=12, height=14, dpi=500)

fig3_diff
```

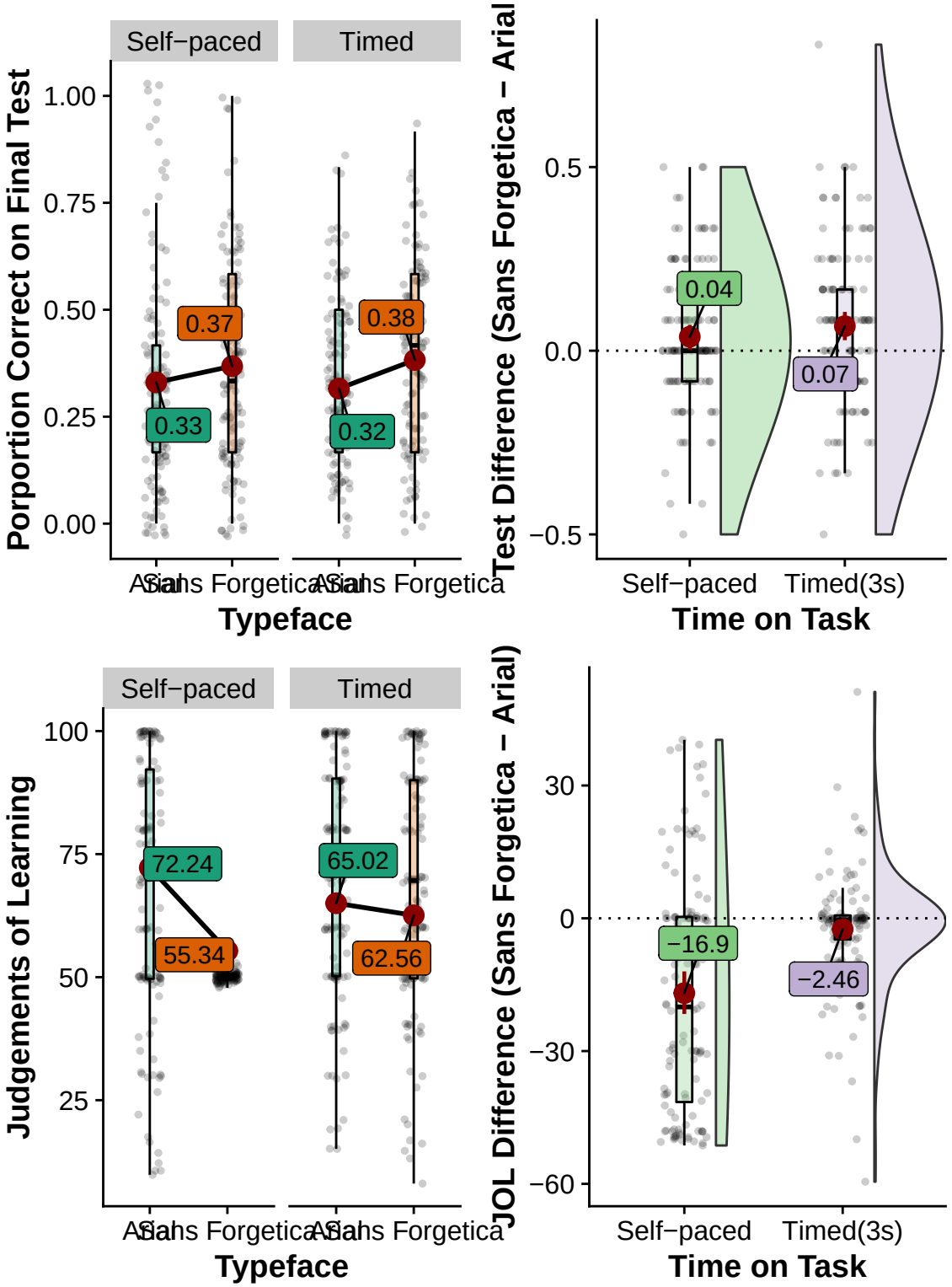

Figure 5: Raincloud plots (Allen et al., 2019) depicting raw data (dots), box plots, and half violin kernel desntiy plots, with mean (red dot) and within-participant 95 CIs. Cued recall accuracy as a function of Time on task for Experiment 3.