Linear mixed model analyses of N400 in single trials

Rstudio Setup

```
knitr::opts_chunk$set(echo = TRUE)
#knitr::opts_knit$set(root.dir = dirname(rstudioapi::getSourceEditorContext()$path)) # set root dir to f
```

Load required libraries & define colours for plotting

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(lmerTest)
```

```
## Loading required package: lme4

## Loading required package: Matrix

##
## Attaching package: 'lmerTest'

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

## The following object is masked from 'package:stats':
##
##     step
```

```
library(ggpubr)
```

```
## Loading required package: ggplot2

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

```
library(reshape2)
cvals <- c('BS' = 'coral1', 'PE' = 'cornflowerblue', 'cat_switch' = 'mediumseagreen')
```

Data loading and cleaning

```
# Load data and clean it
eegdat <- read.csv('../data/roi_300-500ms.csv')
```

```
surp_dat <- read.csv('../outputs/surprise.csv')
dat <- merge(eegdat, surp_dat, all=FALSE)

ntot <- nrow(dat[dat$tau == 1, ])
print(c(ntot, 'trials in total'))
```

```
## [1] "118747"          "trials in total"
```

```
nna <- sum(is.na(dat[dat$tau == 1, ]))
dat <- na.omit(dat)
print(c(nna, 'NA trials removed'))
```

```
## [1] "0"                 "NA trials removed"
```

```
nrem <- sum(dat[dat$tau == 1, ]$badseg)
dat <- dat[dat$badseg == 0,]
print(c(nrem, 'bad trials removed'))
```

```
## [1] "1091"               "bad trials removed"
```

```
nout <- nrow(dat[(dat$tau == 1) & (abs(dat$N400) > 75), ])
dat <- dat[abs(dat$N400) < 75, ] # remove outlier trials
print(c(nout, 'high amplitude trials removed'))
```

```
## [1] "224"                           "high amplitude trials removed"
```

```
print(c(nrow(dat[dat$tau ==1,]), 'trials remaining for analyses'))
```

```
## [1] "117432"                        "trials remaining for analyses"
```

```
dat$word_no <- as.factor(dat$word_no)

# remove unnecessary columns
dat <- select(dat, -c(badseg, X))
```

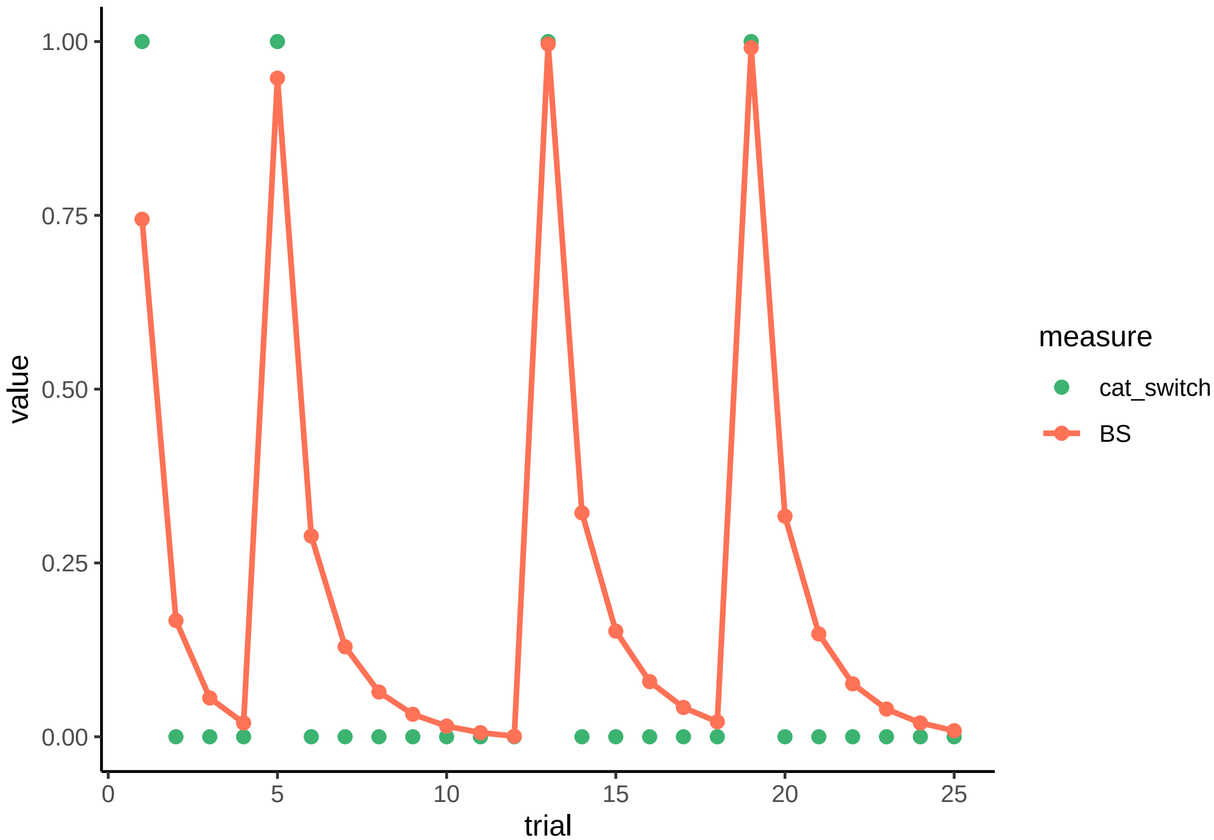
Normalize by max-min scaling, grouped by subject and forgetting parameter.

```
maxmin <- function(x) (x-min(x))/(max(x)-min(x))
dat <- (dat %>% group_by(Subject, tau) %>% mutate(BS = maxmin(BS)))
dat <- (dat %>% group_by(Subject, tau) %>% mutate(PE = maxmin(PE)))
```

Visualisations of semantic surprise & memory decay

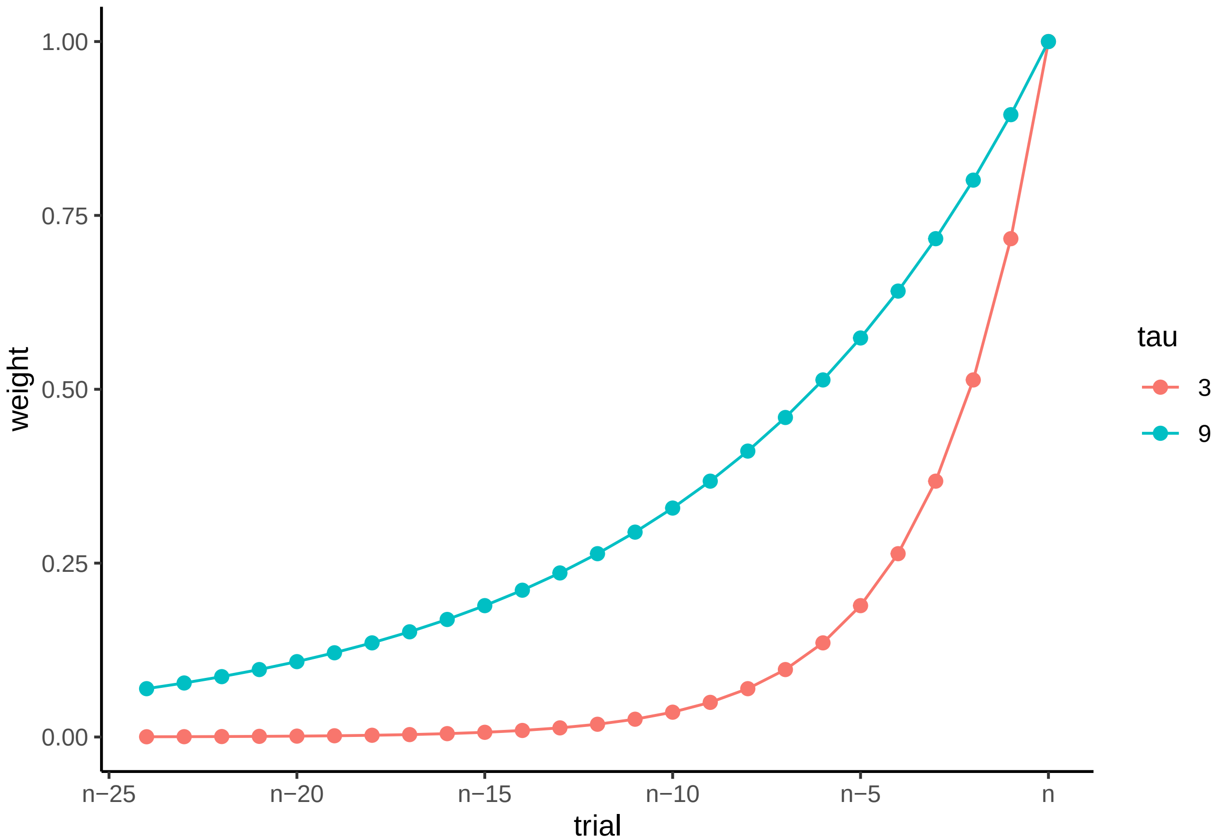
Let's have a look at how the Bayesian surprise evolves over a sequence of 25 trials (category switch measure plotted for comparison).

```
sdat <- dat[dat$Subject == 0,]
sdat <- sdat[sdat$seg <= 25,]
sdat <- sdat[sdat$tau == 3,]
sdat$tau <- as.factor(sdat$tau)

sdat <- rename(sdat, 'trial' = 'seg')
long <- melt(sdat, id.vars = setdiff(names(sdat), c('BS', 'cat_switch')), variable.name = "measure")

plt1 <- ggplot(long, aes(x = trial, y = value, col = measure)) + geom_point(size=2) + geom_line(size=1,

plt1
```



Plot exponential memory decay function for different values of tau.

```
# compute memory filter for a small tau
memfilt <- 25 - 1:25
tau <- 3
weight <- exp(-memfilt/tau)
trial <- 1:25
wind <- data.frame(-memfilt, weight, tau)

tau <- 9 # do the same for a bigger tau
weight <- exp(-memfilt/tau)
wpe <- data.frame(-memfilt, weight, tau)
xbrks <- c(-25, -20, -15, -10, -5)

w2 <- rbind(wind,wpe)
w2$tau <- as.factor(w2$tau)
w2 <- rename(w2, 'trial' = 'X.memfilt')

plt_wind <- ggplot(w2, aes(x = trial, y = weight, color = tau)) + geom_line() + geom_point(size=2) + th
plt_wind
```



Mixed linear models

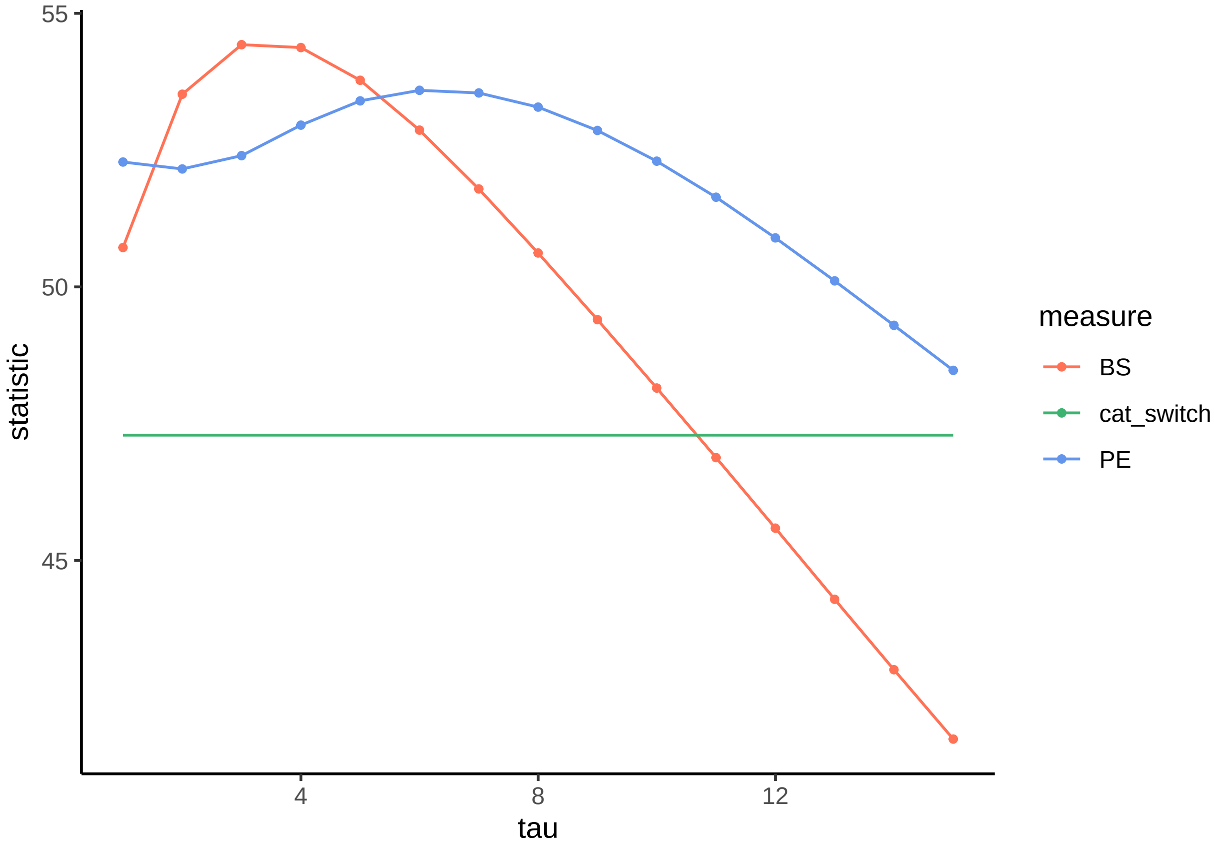
Compute mixed linear models for each measure and value of forgetting parameter (tau).

```
stats_BS <- dat %>%
  group_by(tau) %>%
  group_modify(~ broom::tidy(anova(lmer(scale(N400) ~ BS + (1|Subject) + (1|word_no), data=.x))))

stats_PE <- dat %>%
  group_by(tau) %>%
  group_modify(~ broom::tidy(anova(lmer(scale(N400) ~ PE + (1|Subject) + (1|word_no), data=.x))))

stats_switch <- dat %>%
  group_by(tau) %>%
  group_modify(~ broom::tidy(anova(lmer(scale(N400) ~ cat_switch + (1|Subject) + (1|word_no), data=.x))

stats_BS$param <- 'BS'
stats_PE$param <- 'PE'
stats_switch$param <- 'cat_switch'
allstats <- do.call(rbind, list(stats_BS, stats_PE, stats_switch))
```

Plot the ANOVA F-statistic as function of forgetting parameter (tau)

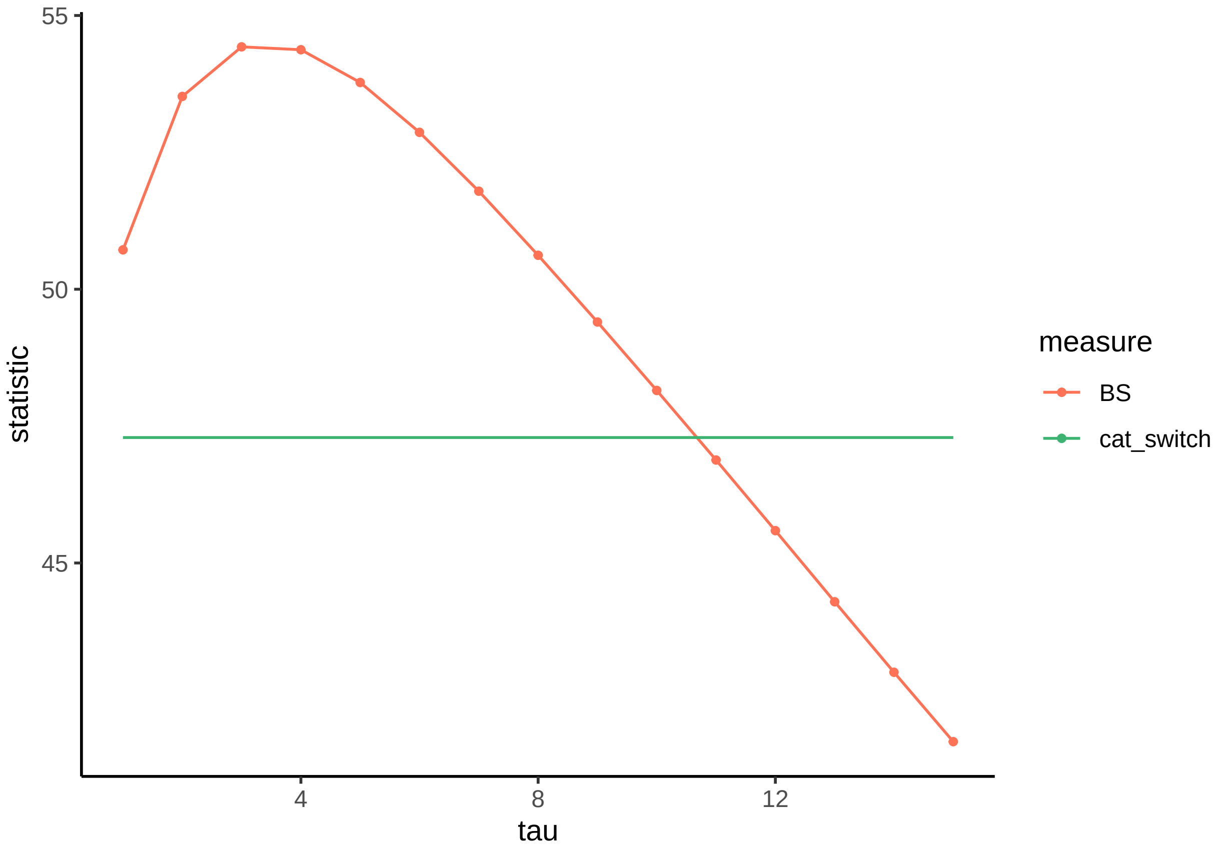
```
allstats$p.value <- p.adjust(allstats$p.value, method='BH')
allstats$significant <- allstats$p.value < 0.05

highlight_pnts <- allstats[allstats$term %in% c('BS', 'PE'),]
```

```
plt <- allstats %>%
  ggplot(aes(x=tau, y=statistic, colour=param)) + geom_line(data=allstats)  + theme_bw() +
  geom_point(data=highlight_pnts, size = 1) + scale_color_manual(values=cvals) +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_
  labs(colour = 'measure')
plt
```



Same plot but excluding PE (as in main text):

```
as <- allstats[allstats$term %in% c('BS', 'cat_switch'),]
highlight_pnts <- as[as$term == 'BS',]

plt_bs_cs <- as %>%
  ggplot(aes(x=tau, y=statistic, colour=param)) + geom_line(data=as)  + theme_bw() +
  geom_point(data=highlight_pnts, size = 1) + scale_color_manual(values=cvals) +
  theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_
  labs(colour = 'measure')
plt_bs_cs
```



Investigate the correlation between Bayesian Surprise and Prediction error for their respective best values of tau.

```
best.tau <- allstats$tau[which.max(allstats$statistic)]

pe_stats <- allstats[allstats$param == 'PE',]
best.tau.PE <- pe_stats$tau[which.max(pe_stats$statistic)]

print(paste('correlation for tau =', best.tau))
```

```
## [1] "correlation for tau = 3"
```

```
sdat <- dat[dat$tau == best.tau,]
cor(sdat[, c('BS','PE')])
```

```
##           BS        PE
## BS 1.0000000 0.9775502
## PE 0.9775502 1.0000000
```

```
print(paste('correlation for tau =', best.tau.PE))
```

```
## [1] "correlation for tau = 6"
```

```
sdat <- dat[dat$tau == best.tau.PE,]
cor(sdat[, c('BS','PE')])
```

```
##           BS        PE
## BS 1.0000000 0.9645834
## PE 0.9645834 1.0000000
```

```r
# save values for best tau to file for further analyses in Python
best.taus <- data.frame('BS' = best.tau, 'PE' = best.tau.PE)
write.csv(best.taus, '../outputs/best_taus.csv')
```

The correlation is almost perfect.

Model comparison

Since the models are non-nested, base model comparison on AIC values.

```r
# compute BS and PE mixed linear models for best performing tau
sdat <- dat[dat$tau == best.tau,]
modBS <- lmer(scale(N400) ~ BS + (1|Subject) + (1|word_no), data=sdat, REML=FALSE)
sdat <- dat[dat$tau == best.tau.PE,]
modPE <- lmer(scale(N400) ~ PE + (1|Subject) + (1|word_no), data=sdat, REML=FALSE)
modCS <- lmer(scale(N400) ~ cat_switch + (1|Subject) + (1|word_no), data=sdat, REML=FALSE)

# compare AIC, including PE
anova.bp <- anova(modBS, modPE, modCS)
# delta AIC
anova.bp['dAIC'] <- anova.bp['AIC']- min(anova.bp['AIC'])
# weighted AIC
anova.bp['wAIC'] = round(exp(-0.5 * anova.bp['dAIC'])/sum(exp(-0.5 * anova.bp['dAIC'])), 4)
anova.bp
```

```
## Data: sdat
## Models:
## modBS: scale(N400) ~ BS + (1 | Subject) + (1 | word_no)
## modPE: scale(N400) ~ PE + (1 | Subject) + (1 | word_no)
## modCS: scale(N400) ~ cat_switch + (1 | Subject) + (1 | word_no)
##       npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)   dAIC
## modBS    5 330932 330980 -165461   330922                      0.0000
## modPE    5 330933 330981 -165461   330923     0  0             0.8354
## modCS    5 330939 330987 -165465   330929     0  0             7.1328
##         wAIC
## modBS 0.5928
## modPE 0.3904
## modCS 0.0168
```

```r
# compare AIC, excluding PE
anova.bp <- anova(modBS, modCS)
# delta AIC
anova.bp['dAIC'] <- anova.bp['AIC']- min(anova.bp['AIC'])
# weighted AIC
anova.bp['wAIC'] = round(exp(-0.5 * anova.bp['dAIC'])/sum(exp(-0.5 * anova.bp['dAIC'])), 4)
anova.bp
```

```
## Data: sdat
## Models:
## modBS: scale(N400) ~ BS + (1 | Subject) + (1 | word_no)
## modCS: scale(N400) ~ cat_switch + (1 | Subject) + (1 | word_no)
##       npar    AIC    BIC  logLik deviance Chisq Df Pr(>Chisq)   dAIC
## modBS    5 330932 330980 -165461   330922                      0.0000
## modCS    5 330939 330987 -165465   330929     0  0             7.1328
##         wAIC
```

```
## modBS 0.9725
## modCS 0.0275
```