Supplement for Sense, Behrens, Meijer & Van Rijn (2015)

compiled by: Florian Sense

Created: June 30, 2015 — Last update: November 28, 2015

Contents


```
library(Matrix)
library(Rcpp)
```

```
## Warning: package 'Rcpp' was built under R version 3.1.3
```

```
library(lme4)
library(BayesFactor)
library(vioplot)
library(zoo)
```

```
## Warning: package 'zoo' was built under R version 3.1.3
```

```r
library(lmtest)
library(MASS)
```

The study

Download the paper

A link to the published paper (PDF) will be added as soon as it is available.

Abstract

One of the goals of computerized tutoring systems is to optimize the learning of facts. Over a hundred years of memory research have identified two robust effects that can improve such systems: the spacing and the testing effect (Cepeda, Pashler, Vul, Wixted, & Rohrer, 2006; Roediger & Butler, 2011). By making optimal use of both and adjusting the system to the individual learner using cognitive models based on memory theories, such systems consistently outperform traditional methods (Van Rijn, Van Maanen & Van Woudenberg, 2009). This adjustment process is driven by a continuously updated estimate of the rate of forgetting for each item and learner on the basis of the learner's accuracy and response time. In this study, we investigated to what extent these estimates of individual rates of forgetting are stable over time and across different materials, and demonstrate that they are stable over time but not across materials. Even though most theories of human memory assume a single underlying rate of forgetting, we show that, in practice, it makes sense to assume different materials are forgotten at different rates. If a computerized, adaptive fact-learning systems would allow for different rates of forgetting for different materials, it could adapt to individual learners more readily.

Data

Description of (data) files

All data is located in the `data/` folder.

- The raw data for each participant for the learning session has been downloaded from the server and merged into a single file. This is also where the **exclusion criteria** were applied. The process is documented in `merge_model_data.html`.
- The participants that fulfilled the exclusion criteria were saved in `whitelist.txt` so that all other files can cross-reference a single file to check whether participants should be included.
- The data for the tests at the end of each block were collected through Google Forms and were exported to individual `.csv` files that can be found in the `/data` folder. Pre-processing consisted of changing the column names and filtering out participants that did not fulfill the exclusion criteria. The files have the prefix `TEST_*`.
- Demographic information was also collected via a Google Form and exported as `demographics.csv` file.
- Mean alpha values were computed for each participant. The means are based on the final alpha values for all items that have been encountered at least three times during the study session. The mean was aggregated across all items to yield one alpha value (i.e., rate of forgetting) per participant. These values are stored for easy access in `mean_alphas.txt`.

- To make the regression analyses easier, I compiled `data_for_Friederike.csv` which contains the alpha value for each item for each participants. Three additional columns contain information about the *block*, which *session* the block was performed in (i.e., on which day), and what type of material was studied (there were four types: Swahili, flags, maps, and biopsychology facts, coded 1 through 4).

Additionally, the `misc/` folder contains:

- A two-line script that determines the labels used for the blocks and their corresponding colors: `block_colors.R`.
- The OmniGraffle files to create Figures 1 and 2 as well as the exported .pdf files that are in the paper.

Reading in the data

The data is distributed across multiple files and this is a bit messy.

```
# List of subject numbers that should be included:
whitelist <- read.table('data/whitelist.txt')[, 1]

# Read in the demographic information:
demo <- read.csv("data/demographics.csv", header=TRUE, stringsAsFactors=FALSE)
names(demo) <- c('timestamp', 'age', 'nationality', 'gender', 'subj', 'america', 'language')
n.before <- length(demo$subj[demo$subj > 60000])
demo <- demo[which(demo$subj %in% whitelist), ]

# Read in the data:
data <- read.csv("data/MODEL_data.csv", stringsAsFactors=FALSE)

# Read in the test data:
# Make sure to replace spaces with periods. The column names contain periods instead of spaces
# and those are used as a reference to determine whether something is spelled correctly!
vocab1 <- read.csv('data/TEST_vocab1.csv', stringsAsFactors=FALSE)
vocab2 <- read.csv('data/TEST_vocab2.csv', stringsAsFactors=FALSE)
vocab3 <- read.csv('data/TEST_vocab.csv', stringsAsFactors=FALSE)
flags <- read.csv('data/TEST_flags.csv', stringsAsFactors=FALSE)
flags <- lapply(flags, function(X) sub(" ", ".", X))
flags$subj <- as.numeric(flags$subj)
flags <- as.data.frame(flags, stringsAsFactors = FALSE)
maps <- read.csv('data/TEST_maps.csv', stringsAsFactors=FALSE)
maps <- lapply(maps, function(X) sub(" ", ".", X))
maps$subj <- as.numeric(maps$subj)
maps <- as.data.frame(maps, stringsAsFactors = FALSE)
biopsych <- read.csv('data/TEST_biopsych.csv', stringsAsFactors=FALSE)

# store the correct responses from the file's headers:
correct <- cbind(names(vocab1), names(vocab2), names(vocab3), names(flags), names(maps), names(biopsych)

# replace header so `col.names` are the same and `rbind` doesn't complain:
header <- c('timestamp', 'subj', paste('item', 1:25, sep=''), 'block')
names(vocab1) <- names(vocab2) <- names(vocab3) <- names(flags) <- names(maps) <- names(biopsych) <- hea

test <- rbind(vocab1, vocab2, vocab3, flags, maps, biopsych)
 # the header names contain periods instead of spaces and those are used as a reference to figure out w
```

```
# Read in data for Friederike's analyses:
data.fr <- read.csv("data/data_for_Friederike.csv")

# Misc:
source("misc/block_colors.R") # load the colors and the labels for the blocks
```

Demographic information

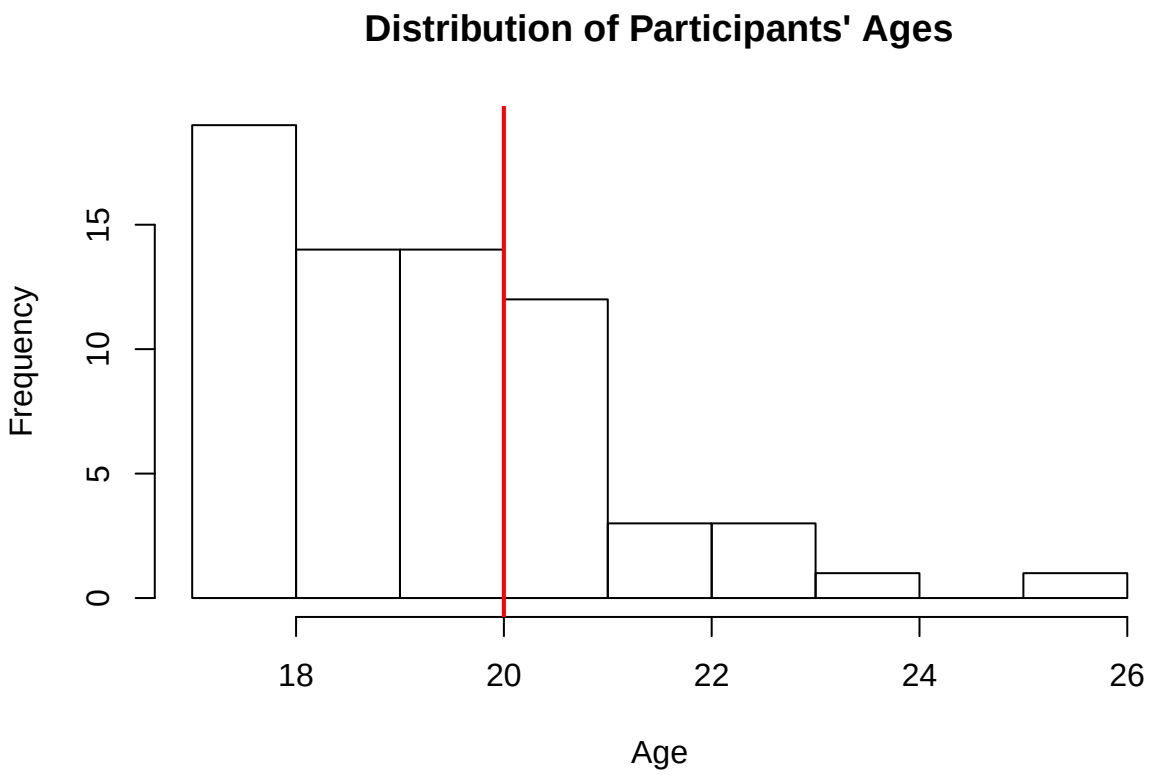
76 participants showed up for the first session. After the exclusion criteria were applied, 67 participants remained in the data set.

Gender

There are 67 participants in the data set of which 17 are male and 50 are female. The median age is 20, with ages between 17, 26 and a standard deviation of 1.74.

Age

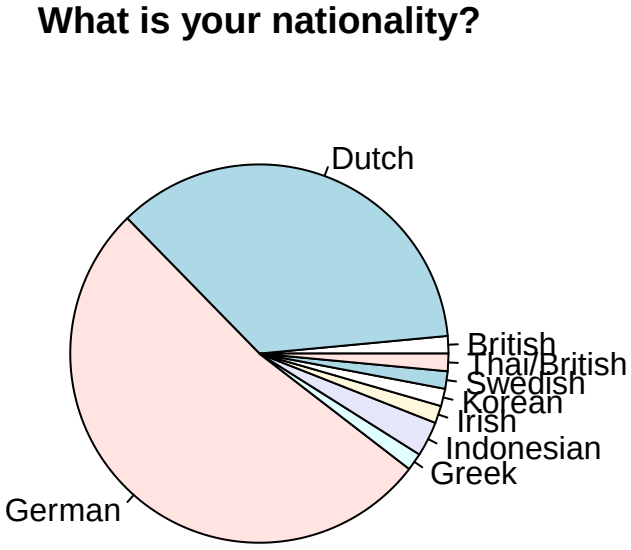
```
hist(demo$age, main='Distribution of Participants\' Ages', xlab='Age')
abline(v=median(demo$age), col="red", lwd=2)
```



Nationality

The nationalities are distributed as follows:

```
pie(table(demo$nationality), main="What is your nationality?")
```

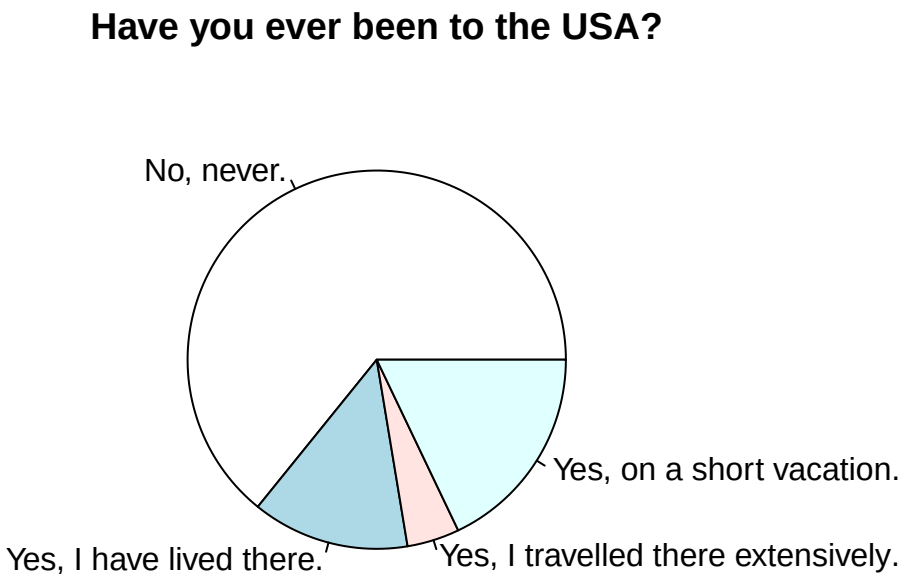


With the biggest group being Germans (52.2%), followed by Dutch citizen (35.8%).

America

We asked them whether they've been to America before to make sure they don't have too much knowledge about the country so we can assume they have no prior knowledge in the `maps` condition. In retrospect, that probably doesn't really matter because the cities are obscure enough and it shouldn't matter even if they happen to know one of them. Here's the distribution of the answers, though:

```
pie(table(demo$america), main='Have you ever been to the USA?')
```



Figures from the paper

Figure 1: Overview of the experimental design

This figure was created in `OmniGraffle Professional` and the file `study_design.graffle` is located in the `/misc` folder.

Figure 2: A graphical representation of the algorithm

This figure was created in `OmniGraffle Professional` and the file `flowchart_algorithm_v2.graffle` is located in the `/misc` folder.

Figure 3: Violin plot of test performance

The proportion correct for each block is computed and the corresponding "violin" is added to the plot:

```
par(las=1, bty='n')
plot(NA, xlim=c(.5, 6.5), ylim=c(.2, 1), axes=F, xlab='', ylab='Proportion Correct')
axis(1, 1:6, blocks)
axis(2, seq(.2, 1, .1))
# title('Percentage Correct on Final Test')

tmp <- apply(vocab1[, 3:27], 1, function(X) mean(tolower(X) == correct[, 1]))
vioplot(tmp, at=1, col=paste(block.col[1], '88', sep=''), add=TRUE, border=NA)
tmp <- apply(vocab2[, 3:27], 1, function(X) mean(tolower(X) == correct[, 2]))
vioplot(tmp, at=2, col=paste(block.col[2], '88', sep=''), add=TRUE, border=NA)
tmp <- apply(vocab3[, 3:27], 1, function(X) mean(tolower(X) == correct[, 3]))
vioplot(tmp, at=3, col=paste(block.col[3], '88', sep=''), add=TRUE, border=NA)
tmp <- apply(flags[, 3:27], 1, function(X) mean(tolower(X) == correct[, 4]))
vioplot(tmp, at=4, col=paste(block.col[4], '88', sep=''), add=TRUE, border=NA)
tmp <- apply(maps[, 3:27], 1, function(X) mean(tolower(X) == correct[, 5]))
vioplot(tmp, at=5, col=paste(block.col[5], '88', sep=''), add=TRUE, border=NA)
tmp <- apply(biopsych[, 3:27], 1, function(X) mean(tolower(X) == correct[, 6]))
vioplot(tmp, at=6, col=paste(block.col[6], '88', sep=''), add=TRUE, border=NA)
```

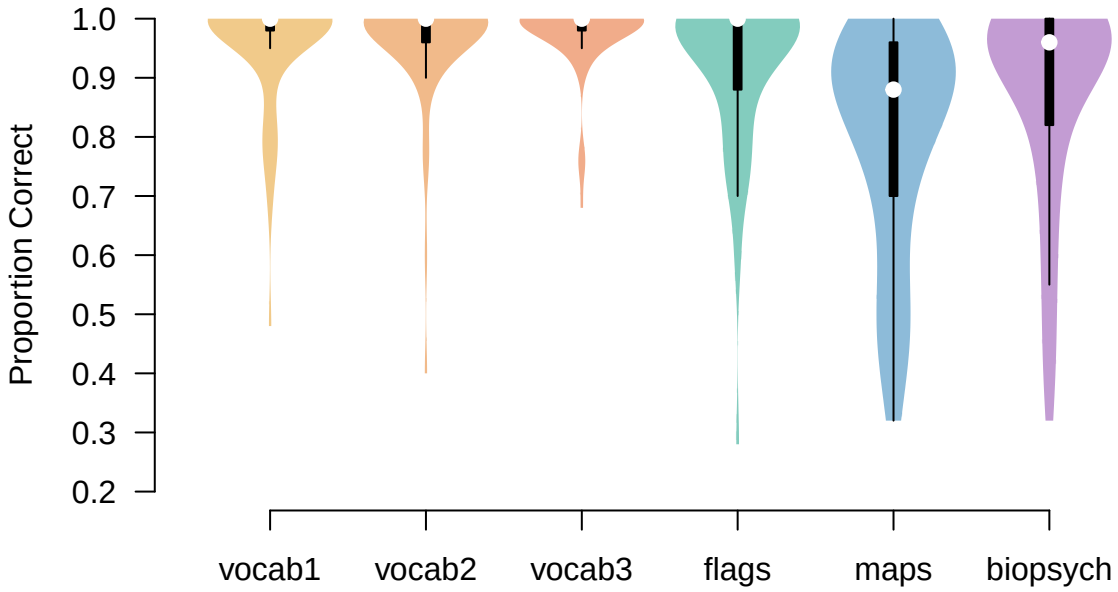


Figure 4: Alpha-test relationship

The function to plot the data:

This function can be used to create plots with varying numbers of bin sizes (or no bins). In the paper, a bin size of 5 was used because it is a good trade-off between preserving the information in the data and providing good readability. Here we are showing the plot with all the raw data along with varying bin sizes:

```
A <- read.table("data/mean_alphas.txt")

par(mfrow=c(2,2))
plot.alpha.test(test, A, use.bins=FALSE)
for(i in c(3, 5, 7)) plot.alpha.test(test, A, n.bin=i)
```

```
## Warning in plot.alpha.test(test, A, n.bin = i): N = 67 which is not a
## multiple of bin size 3 -- therefore 2 NAs have been added to the last bin.

## Warning in plot.alpha.test(test, A, n.bin = i): N = 67 which is not a
## multiple of bin size 5 -- therefore 3 NAs have been added to the last bin.

## Warning in plot.alpha.test(test, A, n.bin = i): N = 67 which is not a
## multiple of bin size 7 -- therefore 3 NAs have been added to the last bin.
```

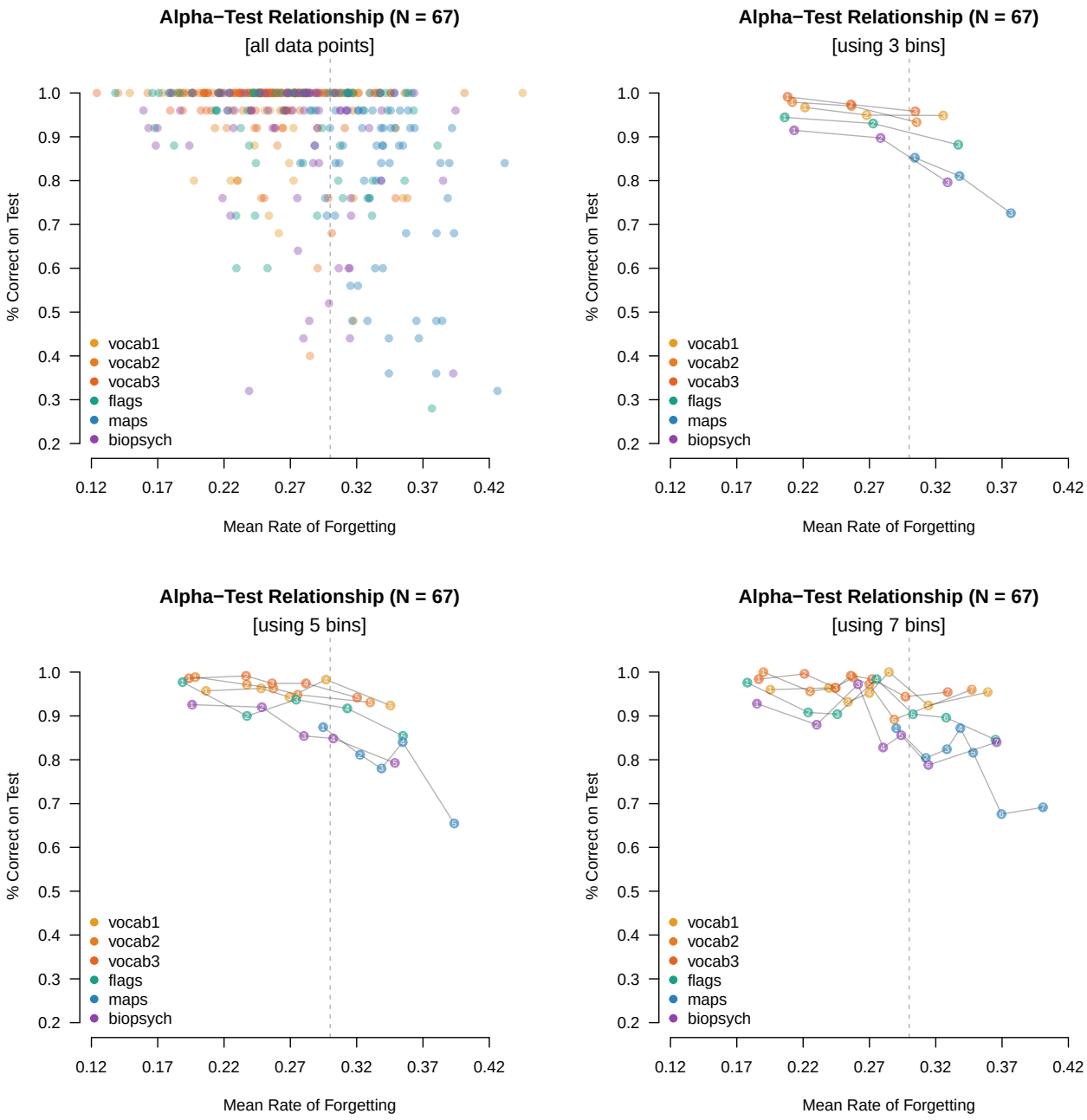
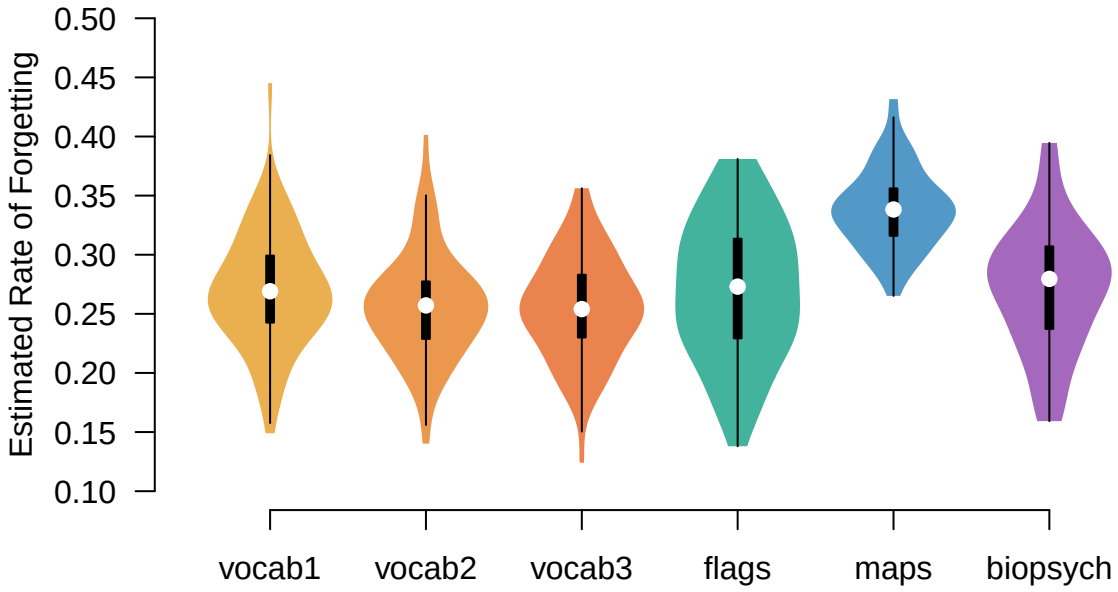


Figure 5: Mean rate of forgetting per participant per condition

Another violin plot, showing estimated rate of forgetting per participant for each of the six conditions.

```
d <- aggregate(alpha ~ block + subj, data.fr, mean)

par(las=1, bty='n')
ylim <- c(.1, .5)
plot(NA, xlim=c(.5, 6.5), ylim=ylim, axes=FALSE, xlab='', ylab='Estimated Rate of Forgetting')

for(i in 1:length(blocks)) {
  vioplot(d$alpha[d$block == blocks[i]], at=i, col=paste(block.col[i], 'CC', sep=''), add=TRUE, border=
```

```
}

axis(1, 1:length(blocks), blocks)
axis(2, seq(ylim[1], ylim[2], .05))  # adjust manually
```



Analyses

Correlations

The correlations were computed for all combinations of blocks:

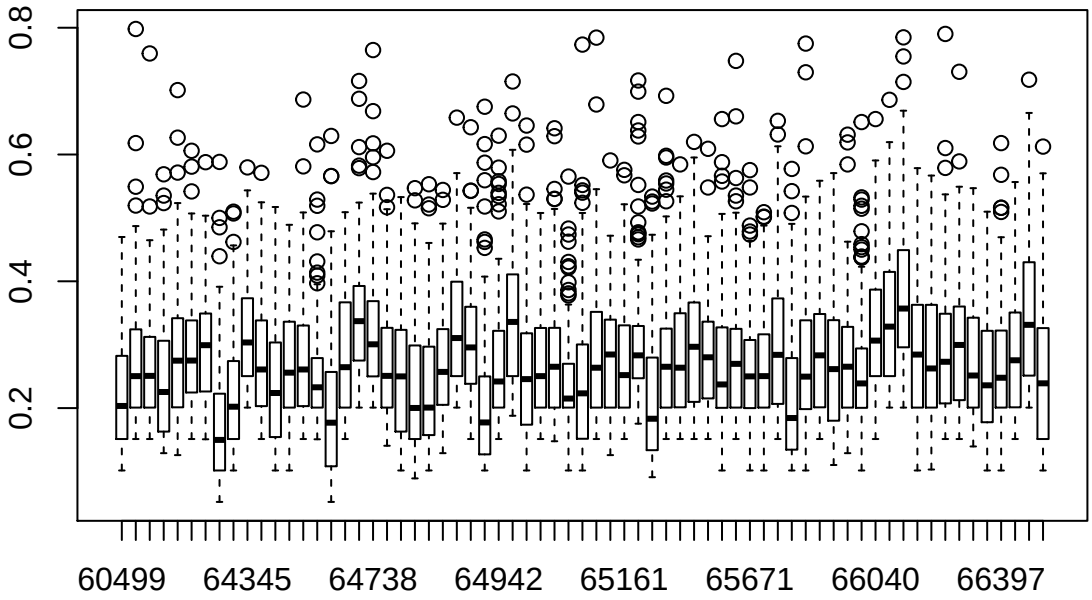
```
## [1] "-----"
## [1] "r = 0.758 with p >= 0 for cor(vocab2, vocab1)"
## [1] "-----"
## [1] "r = 0.771 with p >= 0 for cor(vocab3, vocab1)"
## [1] "r = 0.863 with p >= 0 for cor(vocab3, vocab2)"
## [1] "-----"
## [1] "r = 0.603 with p >= 0 for cor(flags, vocab1)"
## [1] "r = 0.63 with p >= 0 for cor(flags, vocab2)"
## [1] "r = 0.549 with p >= 0 for cor(flags, vocab3)"
## [1] "-----"
## [1] "r = 0.499 with p >= 0 for cor(maps, vocab1)"
## [1] "r = 0.556 with p >= 0 for cor(maps, vocab2)"
## [1] "r = 0.495 with p >= 0 for cor(maps, vocab3)"
## [1] "r = 0.604 with p >= 0 for cor(maps, flags)"
## [1] "-----"
## [1] "r = 0.63 with p >= 0 for cor(biopsych, vocab1)"
## [1] "r = 0.572 with p >= 0 for cor(biopsych, vocab2)"
## [1] "r = 0.534 with p >= 0 for cor(biopsych, vocab3)"
## [1] "r = 0.511 with p >= 0 for cor(biopsych, flags)"
## [1] "r = 0.384 with p >= 0.0013 for cor(biopsych, maps)"
```

The p-values are all tiny and all correlations differ significantly from 0.

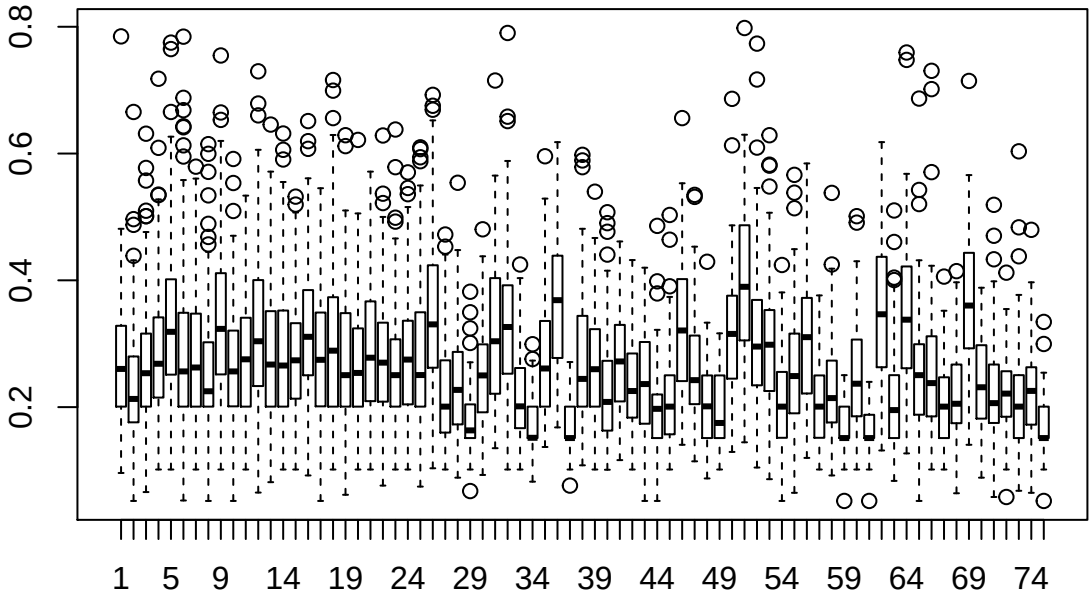
Dummy-coded linear mixed-effects model

This analysis was conducted by **Friederike Behrens**.

```
alphadata <- read.table("data/data_for_Friederike.csv", sep=",", header=TRUE)
attach(alphadata)
boxplot(alpha ~ subj)
```




```
boxplot(alpha ~ item)
```



Great variation between subjects and great variation between items. Both boxplots justify the use of subject and item as random effects (allow for different intercepts per subject and item).

In order to give the interpretation of the regression model a meaningful interpretation, the variables `type` and `session` are recoded from having values from 1-3/4 to values 0-2/3.

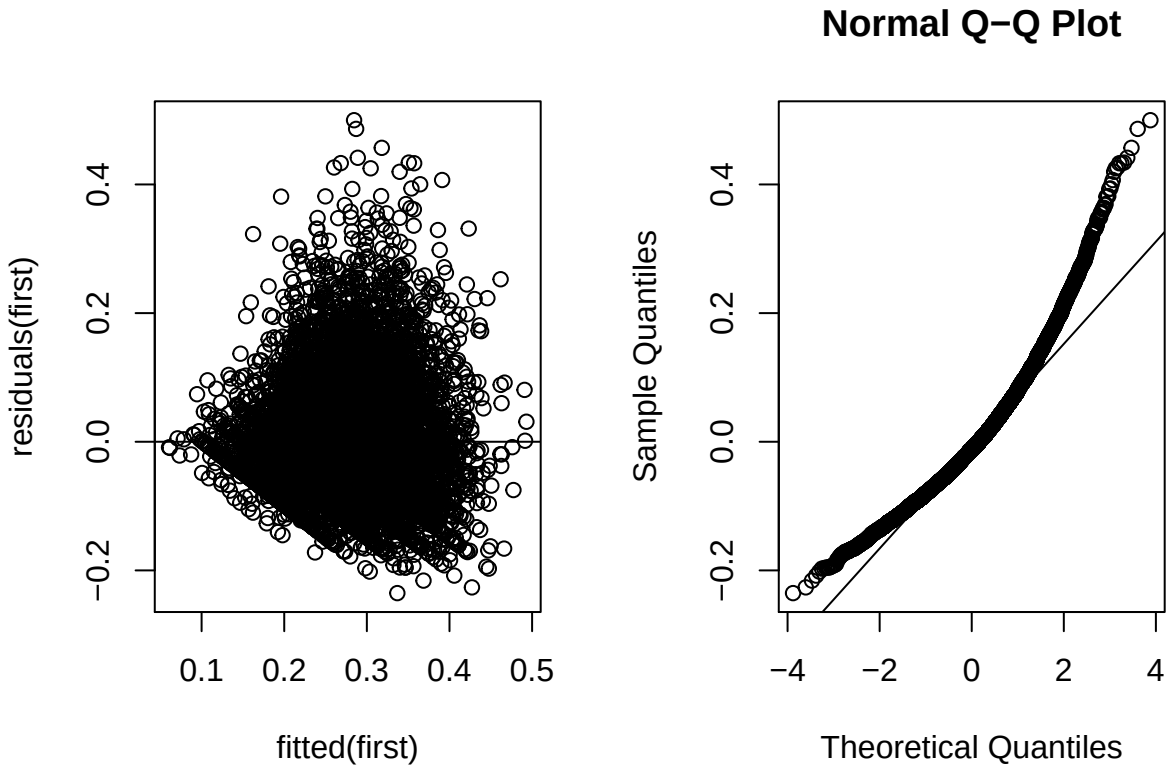
```
sessioncen <- session-1  # centered; make zero meaningful for regression

## dummy coding
# swyn= dummy code with sw= 0 and other tasks(notsw)=1
swyn <- as.numeric( !(type == 1) )
# flci= dummy code with flags= -0.5 and city= 0.5
flci <- 1:length(block) * 0  # all 0's
flci[which(block == 'flags')] <- -.5  # replace at index of "flags"
flci[which(block == 'maps')] <- .5    # replace at index of "maps"/city
```

Fit the first model:

```
first <- lmer(alpha ~ session*swyn + session*flci + (1|subj) + (1|item))
```

```
## fixed-effect model matrix is rank deficient so dropping 1 column / coefficient
```

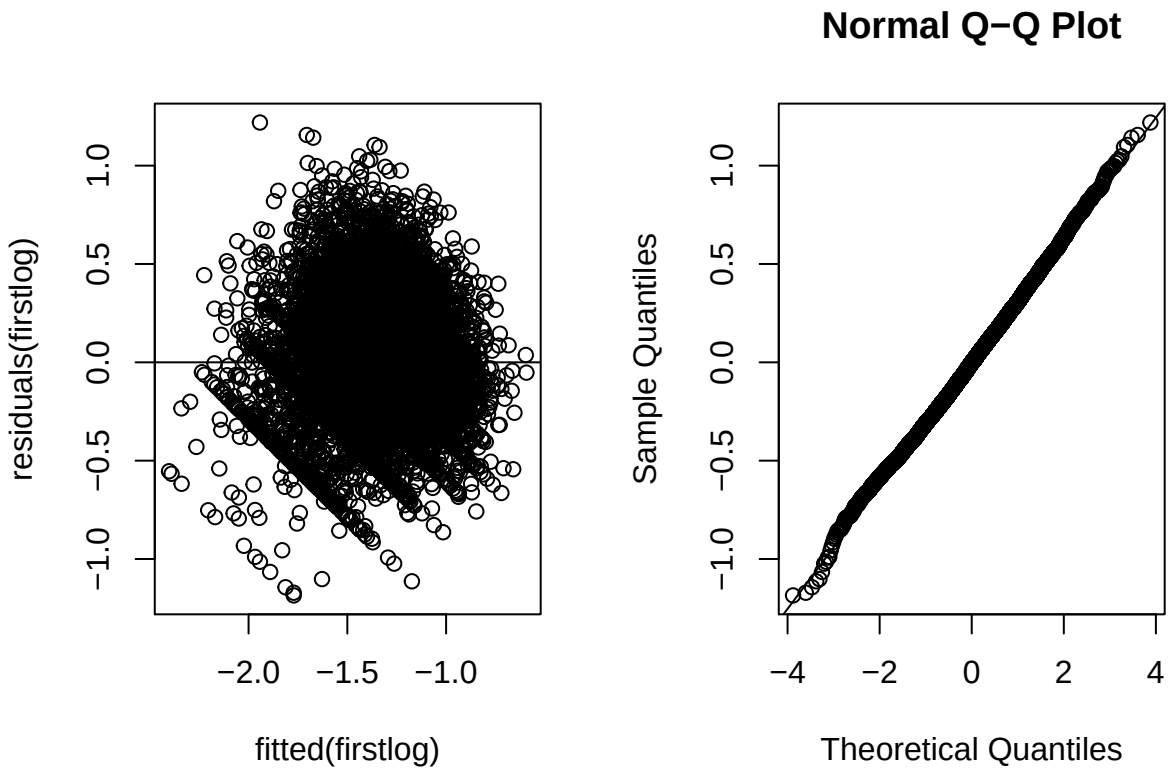
```
# checking assumtpions:
par(mfrow=c(1, 2))
plot(fitted(first),residuals(first))
abline(a=0, b=0)
qqnorm(residuals(first)); qqline(residuals(first))
```



The assumption of linearity is met, because no pattern or shape of a specific non-linear distribution is evident. The assumption of homoscedasticity might be considered violated: the variance for smaller predicted values is smaller compared to the variance for bigger values. Also, the assumption of normality is violated given the big deviations in the qq-plot from the reference line.

One possibility is to log-transform the data:

```r
alphalog <- log(alpha)
firstlog <- lmer(alphalog ~ sessioncen*swyn + flci + (1|subj) + (1|item), data=alphadata)
par(mfrow=c(1, 2))
plot(fitted(firstlog),residuals(firstlog))
abline(a=0, b=0)
qqnorm(residuals(firstlog)); qqline(residuals(firstlog))
```



Both the heteroscedasticity and non-normality is reduced by using the log-transformation of the alpha variable. Consequently, the analysis is conducted using the transformation. Having checked the assumption for using a linear mixed-effects model regression, we can precede with interpreting the proposed model.

```r
summary(firstlog)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: alphalog ~ sessioncen * swyn + flci + (1 | subj) + (1 | item)
##    Data: alphadata
##
## REML criterion at convergence: 5418.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.8003 -0.6908 -0.0093  0.6592  3.9094
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  item     (Intercept) 0.03653  0.1911
##  subj     (Intercept) 0.02514  0.1585
##  Residual             0.09726  0.3119
```

```
## Number of obs: 9602, groups:  item, 75; subj, 67
##
## Fixed effects:
##                 Estimate Std. Error t value
## (Intercept)     -1.39351    0.04053  -34.38
## sessioncen      -0.03581    0.02757   -1.30
## swyn             0.18207    0.01104   16.50
## flci             0.36370    0.01305   27.87
## sessioncen:swyn -0.05124    0.02832   -1.81
##
## Correlation of Fixed Effects:
##             (Intr) sssncn swyn   flci
## sessioncen  -0.680
## swyn        -0.111  0.098
## flci         0.000  0.000  0.313
## sssncn:swyn  0.662 -0.973 -0.229 -0.118
```

Post-hoc t-tests

The analysis revealed that there is a difference between Swahili and non-Swahili conditions (i.e., a sig. effect of `type`). Post-hoc t-tests were performed to check which conditions differ. These basically test the differences between pairs of violins in Figure 5. Since these are post-hoc tests, the critical alpha level should be Bonferroni-correct: alpha is not .05 but .05/6 = .0083.

```
t.test(d$alpha[d$block == "vocab1"], d$alpha[d$block == "vocab2"], paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  d$alpha[d$block == "vocab1"] and d$alpha[d$block == "vocab2"]
## t = 3.3274, df = 66, p-value = 0.001436
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.00539533 0.02158335
## sample estimates:
## mean of the differences
##              0.01348934
```

```
t.test(d$alpha[d$block == "vocab1"], d$alpha[d$block == "vocab3"], paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  d$alpha[d$block == "vocab1"] and d$alpha[d$block == "vocab3"]
## t = 3.8707, df = 66, p-value = 0.0002513
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.007333751 0.022959157
## sample estimates:
## mean of the differences
##              0.01514645
```

```r
t.test(d$alpha[d$block == "vocab2"], d$alpha[d$block == "vocab3"], paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  d$alpha[d$block == "vocab2"] and d$alpha[d$block == "vocab3"]
## t = 0.5784, df = 66, p-value = 0.565
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.004063164  0.007377390
## sample estimates:
## mean of the differences
##             0.001657113
```

```r
t.test(d$alpha[d$block == "flags"], d$alpha[d$block == "maps"], paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  d$alpha[d$block == "flags"] and d$alpha[d$block == "maps"]
## t = -11.9914, df = 66, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.07988669 -0.05708160
## sample estimates:
## mean of the differences
##             -0.06848414
```

```r
t.test(d$alpha[d$block == "flags"], d$alpha[d$block == "biopsych"], paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  d$alpha[d$block == "flags"] and d$alpha[d$block == "biopsych"]
## t = -0.274, df = 66, p-value = 0.785
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01543276  0.01170854
## sample estimates:
## mean of the differences
##            -0.001862111
```

```r
t.test(d$alpha[d$block == "biopsych"], d$alpha[d$block == "maps"], paired = TRUE)
```

```
##
##  Paired t-test
##
## data:  d$alpha[d$block == "biopsych"] and d$alpha[d$block == "maps"]
## t = -10.6706, df = 66, p-value = 5.226e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.07908765 -0.05415641
## sample estimates:
## mean of the differences
##             -0.06662203
```

Additionally, Bayes factor t-tests were performed to see whether the data supports the *equality* of some of the conditions:

```
ttestBF(d$alpha[d$block == "vocab1"], d$alpha[d$block == "vocab2"], paired = TRUE) # ~ 18
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 18.629 <U+00B1>0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

```
ttestBF(d$alpha[d$block == "vocab1"], d$alpha[d$block == "vocab3"], paired = TRUE) # ~ 90
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 90.34678 <U+00B1>0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

```
ttestBF(d$alpha[d$block == "vocab2"], d$alpha[d$block == "vocab3"], paired = TRUE) # 1/BF = ~6.4
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 0.1574038 <U+00B1>0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

```
ttestBF(d$alpha[d$block == "flags"], d$alpha[d$block == "maps"], paired = TRUE) # 1.8 * 10^15
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 1.778431e+15 <U+00B1>0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

```
ttestBF(d$alpha[d$block == "flags"], d$alpha[d$block == "biopsych"], paired = TRUE) # 1/BF = 7.2
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 0.1389866 <U+00B1>0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

```
ttestBF(d$alpha[d$block == "biopsych"], d$alpha[d$block == "maps"], paired = TRUE) # 1.23 * 10^13
```

```
## Bayes factor analysis
## --------------
## [1] Alt., r=0.707 : 1.23222e+13 <U+00B1>0%
##
## Against denominator:
##   Null, mu = 0
## ---
## Bayes factor type: BFoneSample, JZS
```

Backwards regression

This analysis was conducted by **Friederike Behrens**.

The aim of the second analysis is to investigate the predictive power of the forgetting rate based on alpha scores of previous session that consisted of the same or a different task. To do that, the means are calculated for each subject per block (condition).

```
alpha_sw1 <- tapply(alpha[block=="vocab1"], subj[block=="vocab1"], mean)
alpha_sw2 <- tapply(alpha[block=="vocab2"], subj[block=="vocab2"], mean)
alpha_sw3 <- tapply(alpha[block=="vocab3"], subj[block=="vocab3"], mean)
alpha_fl <- tapply(alpha[block=="flags"], subj[block=="flags"], mean)
alpha_ci <- tapply(alpha[block=="maps"], subj[block=="maps"], mean)
```

Next, a backward stepwise regression analysis is conducted.

```
modelstep <- lm(alpha_sw3 ~ alpha_sw2 + alpha_sw1 + alpha_ci + alpha_fl, data=alphadata)
step <- stepAIC(modelstep, direction="backward")
```

```
## Start:  AIC=-510.33
## alpha_sw3 ~ alpha_sw2 + alpha_sw1 + alpha_ci + alpha_fl
##
##             Df Sum of Sq      RSS     AIC
## - alpha_ci   1 0.0000064 0.028404 -512.32
## - alpha_fl   1 0.0001943 0.028591 -511.88
## <none>                   0.028397 -510.33
## - alpha_sw1  1 0.0042526 0.032650 -502.98
## - alpha_sw2  1 0.0211677 0.049565 -475.01
##
```

```
## Step:  AIC=-512.32
## alpha_sw3 ~ alpha_sw2 + alpha_sw1 + alpha_fl
##
##             Df Sum of Sq      RSS     AIC
## - alpha_fl   1 0.0001962 0.028600 -513.86
## <none>                   0.028404 -512.32
## - alpha_sw1  1 0.0042795 0.032683 -504.91
## - alpha_sw2  1 0.0222063 0.050610 -475.62
##
## Step:  AIC=-513.86
## alpha_sw3 ~ alpha_sw2 + alpha_sw1
##
##             Df Sum of Sq      RSS     AIC
## <none>                   0.028600 -513.86
## - alpha_sw1  1 0.0040885 0.032688 -506.90
## - alpha_sw2  1 0.0234198 0.052020 -475.78
```

```
step$anova
```

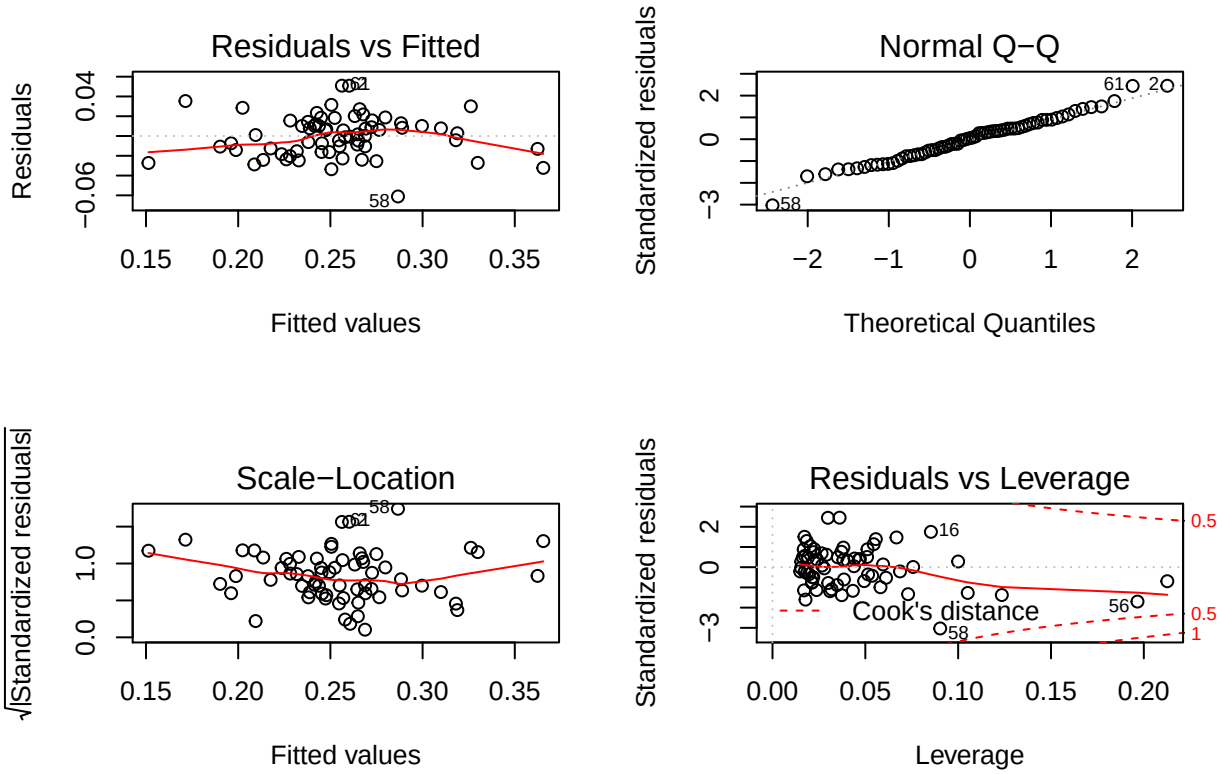
```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## alpha_sw3 ~ alpha_sw2 + alpha_sw1 + alpha_ci + alpha_fl
##
## Final Model:
## alpha_sw3 ~ alpha_sw2 + alpha_sw1
##
##
##         Step Df     Deviance Resid. Df Resid. Dev       AIC
## 1                                   62 0.02839719 -510.3326
## 2 - alpha_ci  1 6.416710e-06        63 0.02840361 -512.3174
## 3 - alpha_fl  1 1.962046e-04        64 0.02859981 -513.8562
```

The ad-hoc hypothesis was that the same and more recent tasks would have more predictive power than another and earlier tasks. Both claims are confirmed by the stewise regression analysis. Only the alpha scores of the same (swahili) task explained a significant proportion of the variance in the alpha scores of the third session of the swahili condition. The tasks of another domain, the flag and city tasks, could not contribute to a better model fit. Moreover, the performance of the more recent swahili task is a stronger predictor than the performance during the first (and therefore earlier) session. This is indicated by the smaller AIC value for the former factor (475.78 versus 506.90), as a smaller AIC indicates a better model fit. Looking at the final model further supports greater contribution of the second compared to the first session indicated by a greater coefficient estimate (0.636 versus 0.244).

```
finalmodel <- lm(alpha_sw3 ~ alpha_sw2 + alpha_sw1, data=alphadata)
```

As a last step, the assumptions for the final model are checked.

```
par(mfrow=c(2,2))
plot(finalmodel)
```



The residuals nicely follow the reference line in the Q-Q plot suggesting a normal distribution of the residuals. Furthermore, the residuals-versus-predicted values plot does not show a clustering or a non-linear shape of the residuals which indicates that the assumptions of homoscedasticity and linearity of residuals are met. Moreover, the residuals-versus-leverage plot does not reveal any particularly influential values.

Taken all together, the stepwise hierarchical regression suggest that the third session of the swahili task can best be predicted by the two preceding sessions of the same task, but not based on tasks from other domains. The total proportion of the variance of the third session of the swahili explained by this model is 77.68% (76.99% adjusted R^2). The final model is as follows:

$$\text{alpha(sw3)} = 0.026 + 0.636 * \text{alpha(sw2)} + 0.244 * \text{alpha(sw1)}$$

Session information

In case there are problems reproducing any of the results/analyses, here's a complete overview of the environment in which this script was run.

```
print(sessionInfo(), locale = FALSE)
```

```
## R version 3.1.2 (2014-10-31)
## Platform: x86_64-apple-darwin10.8.0 (64-bit)
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] MASS_7.3-35       lmtest_0.9-33     zoo_1.7-12
##  [4] vioplot_0.2       sm_2.2-5.4        BayesFactor_0.9.9
```

```
##  [7] coda_0.16-1       lattice_0.20-29   lme4_1.1-7
## [10] Rcpp_0.11.6       Matrix_1.1-4
##
## loaded via a namespace (and not attached):
##  [1] digest_0.6.8    evaluate_0.5.5  formatR_1.0     grid_3.1.2
##  [5] gtools_3.4.1    htmltools_0.2.6 knitr_1.8       minqa_1.2.4
##  [9] mvtnorm_1.0-2   nlme_3.1-118    nloptr_1.0.4    pbapply_1.1-1
## [13] rmarkdown_0.4.2 splines_3.1.2   stringr_0.6.2   tools_3.1.2
## [17] yaml_2.1.13
```