R workshop

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents: <http://rmarkdown.rstudio.com>.

Click the **Knit** button to generate the output document.

Add echo=FALSE or include=FALSE to prevent printing of the R code in output file.

&nbsp; adds space between paragraphs

## Before we start

1/ You need to install R & RStudio and have an internet connection  
2/ Create a folder with the files R\_worshop.rmd and data.csv  
3/ Open R\_worshop.rmd  
4/ Save it as R\_worshop\_YOURNAME.rmd  
5/ Edit that document as we go along (add notes, answer the questions etc …)

## Set up

## define working directory ########## CHANGE !!!!  
setwd("C:/Users/action/Desktop/R\_workshop/")  
  
## define plot theme  
source("plotTheme.R")

## 1) Load R packages

Install all packages you need/load existing packages.

## install new packages  
#install.packages("ggplot2")   
#install.packages("dplyr")   
  
## load packages that are used within this script  
library(ggplot2)  
require(dplyr)

## 2) Import data set

Let’s load some data from a csv file.

data = read.table("data.csv", header=T, sep=",")  
#head(data)

## 3) Prepare data set

Prepare data for your analysis. In R, you need to specify if your data is categorical or continuous to run the analysis properly. By default, R considers variables as continuous.

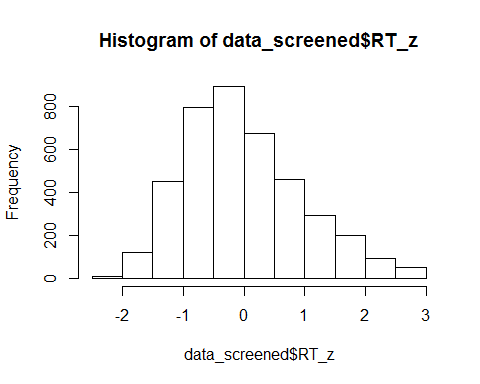
## define categorical variables as factors  
data$trcond <- factor(data$trcond, levels=c(1,2,3), labels=c("Test", "Easy", "Conflict"))  
data$coh <- factor(data$coh, labels=c("low", "high"))  
data$dir <- factor(data$dir, labels=c("left","right"))  
data$id <- factor(data$id)  
  
## exclude error trials  
## %>% specifies that the function 'filter' is applied to 'data'  
data\_correct = data %>% filter(data$error == 0)

Next, we want to remove RTs outside +/- 3 SD. First step, is to calculate z-standardised RTs using the function scale. Because there are big differences between subjects and conditions, we perform this separately for each condition and participants. We add that to the dataset in a new column RT\_z.

## z-standardise RTs for each subject and trial condition separately  
data\_correct = data\_correct %>%  
 group\_by(id, trcond) %>%  
 mutate(RT\_z = as.numeric(scale(RT))) %>%   
 ungroup

**Task 1:**  
a) Exclude RT outliers (outside +/- 3SD) by using the filter function, and call the new data set data\_screened.  
b) Plot a histogram of the screened RTs (hint: you will need the function hist)

## excluding outliers  
data\_screened = data\_correct %>% filter(abs(data\_correct$RT\_z) < 3)  
  
## plot histogramm of RTs  
hist(data\_screened$RT\_z)



## 4) Comparing Means

#### Paired t-test & repeated-measures ANOVA

Aggregate data: The aggregate function allows you to obtain summary statistics (mean/sd/sum/min…) for each id and condition.

## Compute the mean RTs for each id and condition  
RTmean <- aggregate(RT ~ id + trcond,  
 data = data\_screened,   
 mean)  
  
## t-test: compare RTs for the two conditions Test vs. Easy trials  
t.test(RT ~ trcond, data = subset(RTmean, trcond %in% c("Easy", "Test")),  
 paired = TRUE, alternative = "two.sided")

##   
## Paired t-test  
##   
## data: RT by trcond  
## t = 4.0082, df = 16, p-value = 0.001014  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 17.03546 55.28528  
## sample estimates:  
## mean of the differences   
## 36.16037

## one-way ANOVA: compare RTs across all conditions  
results <- aov(RT ~ trcond + Error(id/trcond), data = RTmean)  
  
summary(results)

##   
## Error: id  
## Df Sum Sq Mean Sq F value Pr(>F)  
## Residuals 16 96554 6035   
##   
## Error: id:trcond  
## Df Sum Sq Mean Sq F value Pr(>F)   
## trcond 2 11135 5568 9.106 0.00074 \*\*\*  
## Residuals 32 19565 611   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

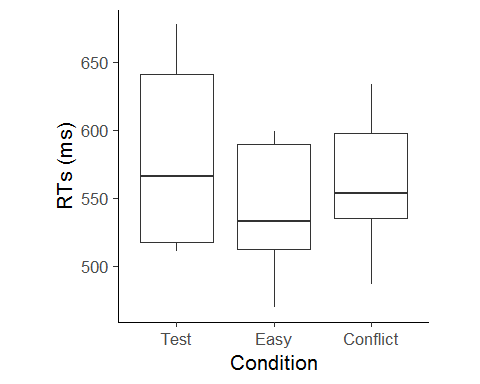
**TASK 2:** RUNNING A 2x2 ANOVA  
a) Average RTs by dot-motion coherence (coh) and dot-motion direction (dir) using the function aggregate. b) Run a repeated-measures 2x2 ANOVA to analyse the main effects and interaction of dot-motion coherence x direction (hint: interaction terms are specified with \*)

## aggregate RTs by coh and dir  
RTmean2 <- aggregate(RT ~ id + coh + dir,  
 data = data\_screened, mean)  
  
## main effects + interactions of coh x dir -> RTs  
resultsInt <- aov(RT ~ coh \* dir + Error(id/(coh\*dir)), data = RTmean2)  
#resultsInt <- aov(RT ~ coh + dir + coh:dir + Error(id/(coh\*dir)), data = RTmean2)

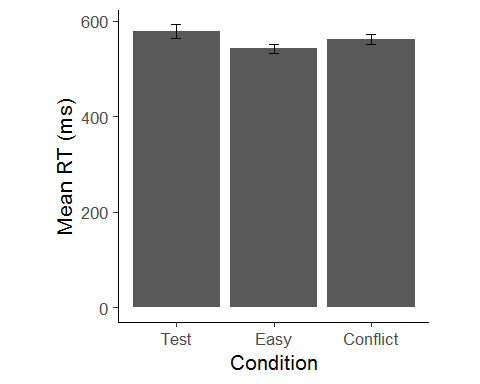
## 6) Plotting Results

#### Barplots & Boxplots

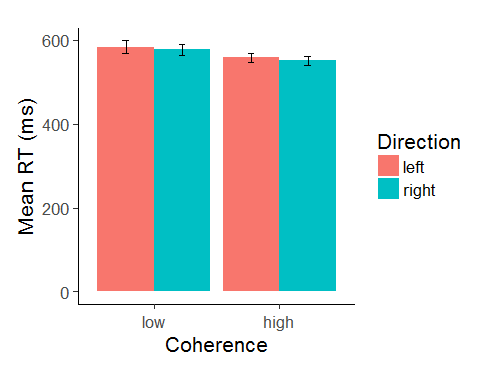
## boxplot of RTs  
ggplot(RTmean, aes(x=trcond, y=RT)) +   
 geom\_boxplot() +  
 labs(x="Condition", y="RTs (ms)") +   
 myTheme()



## barplot of mean RTs  
ggplot(RTmean, aes(x=trcond, y=RT)) +   
 stat\_summary(fun.y=mean, geom="bar") +  
 stat\_summary(fun.data=mean\_se, geom="errorbar", width=.1) +  
 labs(x="Condition", y="Mean RT (ms)") +   
 myTheme()



## barplot of mean RTs: coherence x direction  
pd <- position\_dodge(.9) # move bars from centre so they don't overlap  
ggplot(RTmean2, aes(x=coh, y=RT, fill=dir)) +   
 stat\_summary(fun.y=mean, geom="bar", position=pd) +  
 stat\_summary(fun.data=mean\_se, geom="errorbar", position=pd,width=.1) +  
 #geom\_point(position=pd) +   
 labs(x="Coherence", y="Mean RT (ms)", fill="Direction") +   
 myTheme()



#scale\_fill\_manual(values=c("#006033","#153E7E"))  
 #scale\_fill\_manual(values=c("red","blue"))  
 #scale\_fill\_brewer(palette="Set1") #"Spectral", "Blues" etc.  
  
  
## save plot   
ggsave("RTs.png", scale = 1, width = NA, height = NA, units = "cm", dpi = 300)

## Saving 12.7 x 10.2 cm image

## 7) Regressions & Scatterplots

Correlation between % outcome and SoA judgments

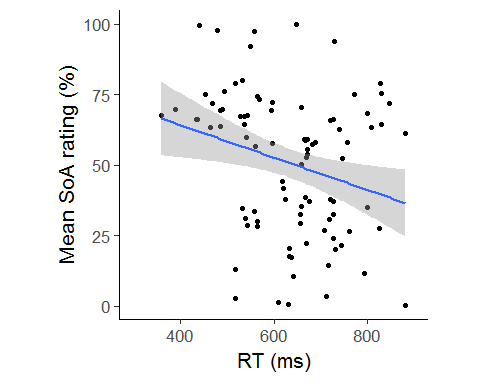
## Select data of participant 1  
dataSub1 <- subset(data\_screened,id %in% "1")  
  
## Regression RT -> SoA   
reg <- lm(SoA ~ RT, data = dataSub1)  
summary(reg)

##   
## Call:  
## lm(formula = SoA ~ RT, data = dataSub1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -54.679 -19.798 4.135 17.697 49.874   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 87.07679 14.08464 6.182 1.87e-08 \*\*\*  
## RT -0.05733 0.02180 -2.630 0.0101 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 24.42 on 89 degrees of freedom  
## (128 observations deleted due to missingness)  
## Multiple R-squared: 0.07212, Adjusted R-squared: 0.06169   
## F-statistic: 6.917 on 1 and 89 DF, p-value: 0.01006

## Plot correlation  
ggplot(dataSub1, aes(x=RT, y=SoA)) +   
 geom\_point() +   
 labs(x="RT (ms)", y="Mean SoA rating (%)") +  
 geom\_smooth(method="lm", se=TRUE, level=0.95) +  
 scale\_x\_continuous(limits=c(300,900)) +   
 myTheme()

## Warning: Removed 128 rows containing non-finite values (stat\_smooth).

## Warning: Removed 128 rows containing missing values (geom\_point).



**TASK 3**:  
a) Run a GLM using the function lm to analyse the effects of both RTs and trcond on SoA.

## Regression RT and trial condition -> SoA   
reg <- lm(SoA ~ RT + trcond, data = dataSub1)  
summary(reg)

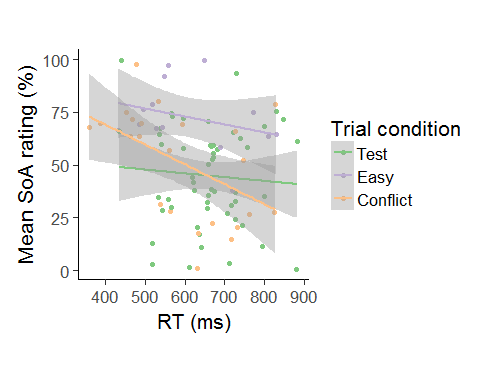
##   
## Call:  
## lm(formula = SoA ~ RT + trcond, data = dataSub1)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -49.036 -16.166 1.883 15.723 52.229   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 76.93525 14.31485 5.375 6.35e-07 \*\*\*  
## RT -0.04865 0.02119 -2.296 0.024092 \*   
## trcondEasy 25.33279 6.96176 3.639 0.000464 \*\*\*  
## trcondConflict 2.87549 5.88955 0.488 0.626612   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 22.98 on 87 degrees of freedom  
## (128 observations deleted due to missingness)  
## Multiple R-squared: 0.1969, Adjusted R-squared: 0.1692   
## F-statistic: 7.109 on 3 and 87 DF, p-value: 0.0002509

Let’s plot the regression line of RTs -> SoA separately for each trial condition.

## Plot correlation  
ggplot(dataSub1, aes(x=RT, y=SoA, color = trcond)) +   
 geom\_point() +   
 labs(x="RT (ms)", y="Mean SoA rating (%)", color = "Trial condition") +  
 geom\_smooth(method="lm", se=TRUE, level=0.95) +  
 myTheme() +   
 #scale\_color\_manual(values=c("red","blue"))  
 scale\_color\_brewer(palette="Accent") #"Spectral", "Blues" etc.

## Warning: Removed 128 rows containing non-finite values (stat\_smooth).

## Warning: Removed 128 rows containing missing values (geom\_point).



## 8) Linear Mixed-Effects Models

Running an LMEM with RTs as a fixed effect and IDs as random intercepts. This model accounts for the fact that participants vary in their overall SoA judgments (each participant has a different intercept). Furthermore, it allows us to include RT as a variable that varies on a trial-by-trial basis within participants (“RTs are nested within IDs”).

## libraries needed for LMEM  
#install.packages("lme4")  
#install.packages("TMB")  
#install.packages("sjPlot")  
  
library(lme4)

## Loading required package: Matrix

library(sjPlot)

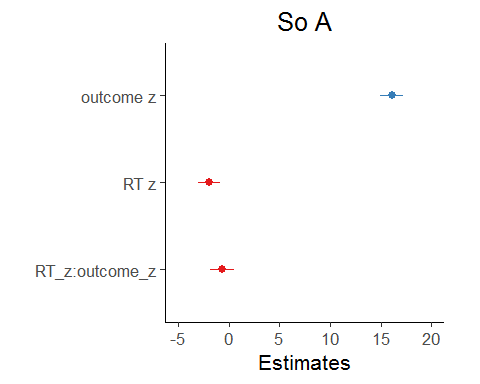
## Install package "strengejacke" from GitHub (`devtools::install\_github("strengejacke/strengejacke")`) to load all sj-packages at once!

data\_screened$outcome\_z = scale(data\_screened$outcome)  
  
## specify model with IDs as random intercept  
lmem <- lmer(SoA ~ RT\_z \* outcome\_z + (1|id),   
 data=data\_screened,  
 control = lmerControl(optCtrl=list(maxfun=1e5)), REML=F)  
  
summary(lmem)

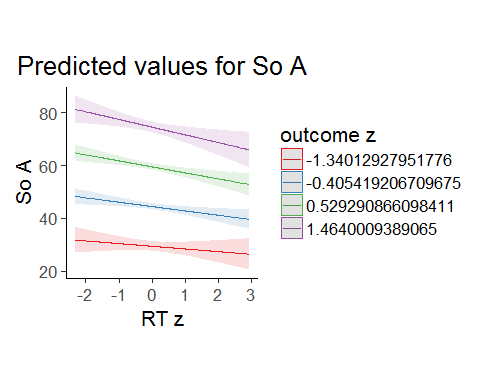
## Linear mixed model fit by maximum likelihood ['lmerMod']  
## Formula: SoA ~ RT\_z \* outcome\_z + (1 | id)  
## Data: data\_screened  
## Control: lmerControl(optCtrl = list(maxfun = 1e+05))  
##   
## AIC BIC logLik deviance df.resid   
## 14198.3 14230.5 -7093.2 14186.3 1578   
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.7357 -0.5241 0.0603 0.6961 3.0291   
##   
## Random effects:  
## Groups Name Variance Std.Dev.  
## id (Intercept) 20.23 4.497   
## Residual 446.09 21.121   
## Number of obs: 1584, groups: id, 17  
##   
## Fixed effects:  
## Estimate Std. Error t value  
## (Intercept) 51.0084 1.2178 41.885  
## RT\_z -1.9278 0.5514 -3.496  
## outcome\_z 16.0767 0.5743 27.992  
## RT\_z:outcome\_z -0.6793 0.5996 -1.133  
##   
## Correlation of Fixed Effects:  
## (Intr) RT\_z otcm\_z  
## RT\_z 0.002   
## outcome\_z 0.022 0.050   
## RT\_z:otcm\_z 0.022 0.032 0.022

## plotting estimates/coefficients of fixed effects with confidence intervals   
plot\_model(lmem, type = "est") + myTheme() # or: sjp.lmer(lmem, type = "fe")

## Computing p-values via Wald-statistics approximation (treating t as Wald z).



## plotting effect of RT on SoA as regression line  
plot\_model(lmem, type = "eff", terms = c("RT\_z", "outcome\_z")) + myTheme()



#sjp.lmer(lmem, type = "ri.slope") # plot slope for each individual  
  
  
## check assumptions of lmem by plotting diagnostics  
#plot\_model(lmem, type = "diag")