SALAD Analyses in R

Lara Wieland

28 April, 2020

# Introduction

This is the analysis of the **main effects of the operant task**.

## Load packages

rm(list = ls())   
  
if (!require(ggplot2)) install.packages("ggplot2")

## Loading required package: ggplot2

if (!require(ggplot2)) install.packages("readxl")  
if (!require(tidyverse)) install.packages("tidyverse")

## Loading required package: tidyverse

## ── Attaching packages ─────────────────────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ tibble 3.0.0 ✓ dplyr 0.8.5  
## ✓ tidyr 1.0.2 ✓ stringr 1.4.0  
## ✓ readr 1.3.1 ✓ forcats 0.5.0  
## ✓ purrr 0.3.3

## ── Conflicts ────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

if (!require(psych)) install.packages("psych")

## Loading required package: psych

##   
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':  
##   
## %+%, alpha

if (!require(reshape2)) install.packages("reshape2")

## Loading required package: reshape2

##   
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':  
##   
## smiths

if (!require(car)) install.packages("car")

## Loading required package: car

## Loading required package: carData

##   
## Attaching package: 'car'

## The following object is masked from 'package:psych':  
##   
## logit

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following object is masked from 'package:purrr':  
##   
## some

if (!require(ez)) install.packages("ez")

## Loading required package: ez

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

if (!require(strex)) install.packages("strex")

## Loading required package: strex

if (!require(nlme)) install.packages("nlme")

## Loading required package: nlme

##   
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':  
##   
## collapse

library("ggplot2")  
library("readxl")  
library("tidyverse")  
library("psych")  
library("car")  
library("reshape2")  
library("ez")  
library("strex")  
library("nlme")

## Set directory and import data

# set path  
setwd('/Users/larawieland/Documents/ECN/Promotion/SALAD/data\_sum')  
  
# import data, set missings (999) to NA and turn into tibble  
data\_imported <- read\_csv2("operant\_prob\_summ\_wo\_excluded.csv",na = c("999", "NA"))

## Using ',' as decimal and '.' as grouping mark. Use read\_delim() for more control.

## Parsed with column specification:  
## cols(  
## .default = col\_character(),  
## group = col\_double(),  
## order = col\_double(),  
## excluded = col\_double(),  
## age = col\_double(),  
## weight = col\_number(),  
## height = col\_double(),  
## education = col\_double(),  
## tmt\_a = col\_double(),  
## tmt\_b = col\_double(),  
## dsst = col\_double(),  
## num\_forward = col\_double(),  
## num\_backward = col\_double(),  
## IQ\_WST = col\_double()  
## )

## See spec(...) for full column specifications.

dat <- as\_tibble(data\_imported)  
dat\_complete <- na.omit(dat)   
rm(data\_imported)  
  
# remove outliers (according to < 55% p\_correct in all trials or 1/3 phases)  
  
dat.outlier.T1 <- dat %>% filter(p\_correct\_T1 < .55)  
dat.outlier.T2 <- dat %>% filter(p\_correct\_T2 < .55)

## Play with dplyr functions to sort, subset and arrange

# subset data for order (A=CT-ST,B=ST-CT)  
dat.order.A <- subset(dat, order == "1")  
dat.order.B <- subset(dat, order == "2")  
  
# the following is just playing around:  
# arrange by age in descending order  
data\_age\_sorted <- dat %>% arrange(desc(age))  
  
# change variable height from cm to m  
dat.height\_in\_m <- dat %>% mutate(height = height/100)  
  
rm(data\_age\_sorted)  
rm(dat.height\_in\_m)

## Demographics and NP - Descriptive Stats and T-Tests between groups

# These reproduced Zsuzsi's results: significant differences between groups in age and education, IQ\_WST, num\_forward, num\_backward and DSST  
  
by(dat$age, dat$group, mean)

## dat$group: 1  
## [1] 26.93103  
## ------------------------------------------------------------   
## dat$group: 2  
## [1] 37.33333

# by(dat$age, dat$group, sd)  
t.test(age ~ group, data=dat)

##   
## Welch Two Sample t-test  
##   
## data: age by group  
## t = -4.5075, df = 38.638, p-value = 5.932e-05  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -15.071576 -5.733022  
## sample estimates:  
## mean in group 1 mean in group 2   
## 26.93103 37.33333

by(dat$education, dat$group, mean)

## dat$group: 1  
## [1] 2.689655  
## ------------------------------------------------------------   
## dat$group: 2  
## [1] 1.555556

# by(dat$education, dat$group, sd)  
t.test(education ~ group, data=dat)

##   
## Welch Two Sample t-test  
##   
## data: education by group  
## t = 7.1848, df = 53.958, p-value = 2.073e-09  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.8176293 1.4505700  
## sample estimates:  
## mean in group 1 mean in group 2   
## 2.689655 1.555556

by(dat$weight, dat$group, mean)

## dat$group: 1  
## [1] 77.93103  
## ------------------------------------------------------------   
## dat$group: 2  
## [1] 108.8889

# by(dat$weight, dat$group, sd)  
t.test(weight ~ group, data=dat)

##   
## Welch Two Sample t-test  
##   
## data: weight by group  
## t = -1.1164, df = 26.224, p-value = 0.2744  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -87.93250 26.01679  
## sample estimates:  
## mean in group 1 mean in group 2   
## 77.93103 108.88889

# for IQ with dat\_complete due to some NAs  
by(dat\_complete$IQ\_WST, dat\_complete$group, mean)

## dat\_complete$group: 1  
## [1] 104.1786  
## ------------------------------------------------------------   
## dat\_complete$group: 2  
## [1] 98.05

# by(dat$IQ\_WST, dat$group, sd)  
t.test(IQ\_WST ~ group, data=dat\_complete)

##   
## Welch Two Sample t-test  
##   
## data: IQ\_WST by group  
## t = 2.2789, df = 44.622, p-value = 0.02751  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.7108206 11.5463223  
## sample estimates:  
## mean in group 1 mean in group 2   
## 104.1786 98.0500

by(dat$num\_forward, dat$group, mean)

## dat$group: 1  
## [1] 10.24138  
## ------------------------------------------------------------   
## dat$group: 2  
## [1] 8.814815

# by(dat$num\_forward, dat$group, sd)  
t.test(num\_forward ~ group, data=dat)

##   
## Welch Two Sample t-test  
##   
## data: num\_forward by group  
## t = 2.535, df = 52.459, p-value = 0.01426  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.2975826 2.5555464  
## sample estimates:  
## mean in group 1 mean in group 2   
## 10.241379 8.814815

by(dat$num\_backward, dat$group, mean)

## dat$group: 1  
## [1] 7.586207  
## ------------------------------------------------------------   
## dat$group: 2  
## [1] 6.333333

# by(dat$num\_backward, dat$group, sd)  
t.test(num\_backward ~ group, data=dat)

##   
## Welch Two Sample t-test  
##   
## data: num\_backward by group  
## t = 2.0865, df = 53.942, p-value = 0.04168  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 0.04897446 2.45677267  
## sample estimates:  
## mean in group 1 mean in group 2   
## 7.586207 6.333333

by(dat$tmt\_a, dat$group, mean)

## dat$group: 1  
## [1] 29.18  
## ------------------------------------------------------------   
## dat$group: 2  
## [1] 27.2763

# by(dat$tmt\_a, dat$group, sd)  
t.test(tmt\_a ~ group, data=dat)

##   
## Welch Two Sample t-test  
##   
## data: tmt\_a by group  
## t = 0.72092, df = 46.859, p-value = 0.4745  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -3.409031 7.216438  
## sample estimates:  
## mean in group 1 mean in group 2   
## 29.1800 27.2763

by(dat$tmt\_b, dat$group, mean)

## dat$group: 1  
## [1] 62.81276  
## ------------------------------------------------------------   
## dat$group: 2  
## [1] NA

# by(dat$tmt\_b, dat$group, sd)  
t.test(tmt\_b ~ group, data=dat)

##   
## Welch Two Sample t-test  
##   
## data: tmt\_b by group  
## t = -1.358, df = 50.535, p-value = 0.1805  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -24.10974 4.65603  
## sample estimates:  
## mean in group 1 mean in group 2   
## 62.81276 72.53962

by(dat$dsst, dat$group, mean)

## dat$group: 1  
## [1] 80.72414  
## ------------------------------------------------------------   
## dat$group: 2  
## [1] 65.55556

# by(dat$dsst, dat$group, sd)  
t.test(dsst ~ group, data=dat)

##   
## Welch Two Sample t-test  
##   
## data: dsst by group  
## t = 4.9341, df = 52.724, p-value = 8.452e-06  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## 9.001675 21.335490  
## sample estimates:  
## mean in group 1 mean in group 2   
## 80.72414 65.55556

dat %>% summarize(mean\_age = mean(age))

## # A tibble: 1 x 1  
## mean\_age  
## <dbl>  
## 1 31.9

## Preparations for rmANOVA

# as a first attempt tried with p\_correct as dv for rmANOVA with within = volat (pre)  
  
# prepare data first: p\_correct\_T1 needs to be turned into p\_correct\_CT and P\_correct\_T2 into P\_Correct\_S for order = 1(A)  
  
dat.order.CTST <- dat.order.A %>%   
  
rename(mean\_RT\_CT = mean\_RT\_T1) %>%   
rename(p\_correct\_CT = p\_correct\_T1) %>%   
rename(p\_stay\_CT = p\_stay\_T1) %>%   
rename(p\_switch\_CT = p\_switch\_T1) %>%  
rename(p\_win\_stay\_CT = p\_win\_stay\_T1) %>%   
rename(p\_win\_switch\_CT = p\_win\_switch\_T1) %>%   
rename(p\_lose\_stay\_CT = p\_lose\_stay\_T1) %>%   
rename(p\_lose\_switch\_CT = p\_lose\_switch\_T1) %>%   
   
rename(mean\_RT\_ST = mean\_RT\_T2) %>%   
rename(p\_correct\_ST = p\_correct\_T2) %>%   
rename(p\_stay\_ST = p\_stay\_T2) %>%   
rename(p\_switch\_ST = p\_switch\_T2) %>%   
rename(p\_win\_stay\_ST = p\_win\_stay\_T2) %>%   
rename(p\_win\_switch\_ST = p\_win\_switch\_T2) %>%   
rename(p\_lose\_stay\_ST = p\_lose\_stay\_T2) %>%   
rename(p\_lose\_switch\_ST = p\_lose\_switch\_T2) %>%  
   
rename(p\_RT\_pre\_CT = p\_RT\_pre\_T1) %>%   
rename(p\_RT\_rev\_CT = p\_RT\_rev\_T1) %>%   
rename(p\_RT\_post\_CT = p\_RT\_post\_T1) %>%   
rename(p\_correct\_pre\_CT = p\_correct\_pre\_T1) %>%   
rename(p\_correct\_rev\_CT = p\_correct\_rev\_T1) %>%   
rename(p\_correct\_post\_CT = p\_correct\_post\_T1) %>%   
rename(p\_w\_st\_pre\_CT = p\_w\_st\_pre\_T1) %>%   
rename(p\_w\_st\_rev\_CT = p\_w\_st\_rev\_T1) %>%   
rename(p\_w\_st\_post\_CT = p\_w\_st\_post\_T1) %>%   
rename(p\_l\_sw\_pre\_CT = p\_l\_sw\_pre\_T1) %>%   
rename(p\_l\_sw\_rev\_CT = p\_l\_sw\_rev\_T1) %>%   
rename(p\_l\_sw\_post\_CT = p\_l\_sw\_post\_T1) %>%   
   
rename(p\_RT\_pre\_ST = p\_RT\_pre\_T2) %>%   
rename(p\_RT\_rev\_ST = p\_RT\_rev\_T2) %>%   
rename(p\_RT\_post\_ST = p\_RT\_post\_T2) %>%   
rename(p\_correct\_pre\_ST = p\_correct\_pre\_T2) %>%   
rename(p\_correct\_rev\_ST = p\_correct\_rev\_T2) %>%   
rename(p\_correct\_post\_ST = p\_correct\_post\_T2) %>%   
rename(p\_w\_st\_pre\_ST = p\_w\_st\_pre\_T2) %>%   
rename(p\_w\_st\_rev\_ST = p\_w\_st\_rev\_T2) %>%   
rename(p\_w\_st\_post\_ST = p\_w\_st\_post\_T2) %>%   
rename(p\_l\_sw\_pre\_ST = p\_l\_sw\_pre\_T2) %>%   
rename(p\_l\_sw\_rev\_ST = p\_l\_sw\_rev\_T2) %>%   
rename(p\_l\_sw\_post\_ST = p\_l\_sw\_post\_T2)  
  
# and the other way around for order = 2(B)  
  
dat.order.STCT <- dat.order.B %>%   
  
rename(mean\_RT\_CT = mean\_RT\_T2) %>%   
rename(p\_correct\_CT = p\_correct\_T2) %>%   
rename(p\_stay\_CT = p\_stay\_T2) %>%   
rename(p\_switch\_CT = p\_switch\_T2) %>%  
rename(p\_win\_stay\_CT = p\_win\_stay\_T2) %>%   
rename(p\_win\_switch\_CT = p\_win\_switch\_T2) %>%   
rename(p\_lose\_stay\_CT = p\_lose\_stay\_T2) %>%   
rename(p\_lose\_switch\_CT = p\_lose\_switch\_T2) %>%   
   
rename(mean\_RT\_ST = mean\_RT\_T1) %>%   
rename(p\_correct\_ST = p\_correct\_T1) %>%   
rename(p\_stay\_ST = p\_stay\_T1) %>%   
rename(p\_switch\_ST = p\_switch\_T1) %>%   
rename(p\_win\_stay\_ST = p\_win\_stay\_T1) %>%   
rename(p\_win\_switch\_ST = p\_win\_switch\_T1) %>%   
rename(p\_lose\_stay\_ST = p\_lose\_stay\_T1) %>%   
rename(p\_lose\_switch\_ST = p\_lose\_switch\_T1) %>%  
   
rename(p\_RT\_pre\_CT = p\_RT\_pre\_T2) %>%   
rename(p\_RT\_rev\_CT = p\_RT\_rev\_T2) %>%   
rename(p\_RT\_post\_CT = p\_RT\_post\_T2) %>%   
rename(p\_correct\_pre\_CT = p\_correct\_pre\_T2) %>%   
rename(p\_correct\_rev\_CT = p\_correct\_rev\_T2) %>%   
rename(p\_correct\_post\_CT = p\_correct\_post\_T2) %>%   
rename(p\_w\_st\_pre\_CT = p\_w\_st\_pre\_T2) %>%   
rename(p\_w\_st\_rev\_CT = p\_w\_st\_rev\_T2) %>%   
rename(p\_w\_st\_post\_CT = p\_w\_st\_post\_T2) %>%   
rename(p\_l\_sw\_pre\_CT = p\_l\_sw\_pre\_T2) %>%   
rename(p\_l\_sw\_rev\_CT = p\_l\_sw\_rev\_T2) %>%   
rename(p\_l\_sw\_post\_CT = p\_l\_sw\_post\_T2) %>%   
   
rename(p\_RT\_pre\_ST = p\_RT\_pre\_T1) %>%   
rename(p\_RT\_rev\_ST = p\_RT\_rev\_T1) %>%   
rename(p\_RT\_post\_ST = p\_RT\_post\_T1) %>%   
rename(p\_correct\_pre\_ST = p\_correct\_pre\_T1) %>%   
rename(p\_correct\_rev\_ST = p\_correct\_rev\_T1) %>%   
rename(p\_correct\_post\_ST = p\_correct\_post\_T1) %>%   
rename(p\_w\_st\_pre\_ST = p\_w\_st\_pre\_T1) %>%   
rename(p\_w\_st\_rev\_ST = p\_w\_st\_rev\_T1) %>%   
rename(p\_w\_st\_post\_ST = p\_w\_st\_post\_T1) %>%   
rename(p\_l\_sw\_pre\_ST = p\_l\_sw\_pre\_T1) %>%   
rename(p\_l\_sw\_rev\_ST = p\_l\_sw\_rev\_T1) %>%   
rename(p\_l\_sw\_post\_ST = p\_l\_sw\_post\_T1)  
  
# merge them back together and clean WS  
dat\_all <- bind\_rows(dat.order.CTST,dat.order.STCT)  
  
rm(dat.order.CTST)  
rm(dat.order.STCT)  
rm(dat.order.A)  
rm(dat.order.B)

## transfer from wide to long format p\_correct

# transfer them to long format as preparation for rmANOVA  
  
longdat.correct <- melt(dat\_all,  
 # ID variables - all the variables to keep but not split apart on  
 id.vars=c("sub\_id", "group","order"),  
 # The source columns  
 measure.vars=c("p\_correct\_pre\_ST","p\_correct\_rev\_ST","p\_correct\_post\_ST","p\_correct\_pre\_CT","p\_correct\_rev\_CT","p\_correct\_post\_CT"),  
 # Name of the destination column that will identify the original  
 # column that the measurement came from  
 variable.name="phase",  
 value.name="p\_correct"  
)  
  
# use package strex to extract last number from sub\_id and turn it into factor  
longdat.correct$id <- str\_last\_number(longdat.correct$sub\_id)  
longdat.correct$id <- as.factor(longdat.correct$id)  
longdat.correct$p\_correct <- as.numeric(longdat.correct$p\_correct)  
  
longdat.correct <- longdat.correct %>% arrange(id)  
  
longdat.correct <- longdat.correct %>% arrange(phase)  
  
longdat.correct$volat <- gl(6,56,labels=c("pre\_ST","rev\_ST","post\_ST","pre\_CT","rev\_CT","post\_CT"))  
  
levels(longdat.correct$volat)[levels(longdat.correct$volat)=="pre\_ST"] <- "pre\_CT"  
levels(longdat.correct$volat)[levels(longdat.correct$volat)=="rev\_ST"] <- "rev\_CT"  
levels(longdat.correct$volat)[levels(longdat.correct$volat)=="post\_ST"] <- "post\_CT"  
  
longdat.correct <- longdat.correct %>% arrange(id)

## transfer from wide to long format p\_winstay

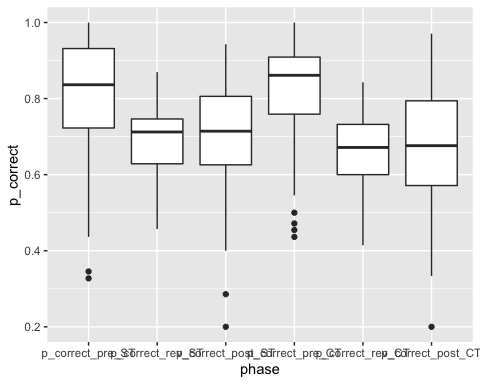
# transfer them to long format as preparation for rmANOVA  
  
longdat.winstay <- melt(dat\_all,  
 # ID variables - all the variables to keep but not split apart on  
 id.vars=c("sub\_id", "group","order"),  
 # The source columns  
 measure.vars=c("p\_w\_st\_pre\_ST","p\_w\_st\_rev\_ST","p\_w\_st\_post\_ST","p\_w\_st\_pre\_CT","p\_w\_st\_rev\_CT","p\_w\_st\_post\_CT"),  
 # Name of the destination column that will identify the original  
 # column that the measurement came from  
 variable.name="phase",  
 value.name="p\_winstay"  
)  
  
# use package strex to extract last number from sub\_id and turn it into factor  
longdat.winstay$id <- str\_last\_number(longdat.winstay$sub\_id)  
longdat.winstay$id <- as.factor(longdat.winstay$id)  
longdat.winstay$p\_winstay <- as.numeric(longdat.winstay$p\_winstay)  
  
longdat.winstay <- longdat.winstay %>% arrange(id)  
  
longdat.winstay <- longdat.winstay %>% arrange(phase)  
  
longdat.winstay$volat <- gl(6,56,labels=c("pre\_ST","rev\_ST","post\_ST","pre\_CT","rev\_CT","post\_CT"))  
  
levels(longdat.winstay$volat)[levels(longdat.winstay$volat)=="pre\_ST"] <- "pre\_CT"  
levels(longdat.winstay$volat)[levels(longdat.winstay$volat)=="rev\_ST"] <- "rev\_CT"  
levels(longdat.winstay$volat)[levels(longdat.winstay$volat)=="post\_ST"] <- "post\_CT"  
  
longdat.winstay <- longdat.winstay %>% arrange(id)

## transfer from wide to long format p\_RT

# transfer them to long format as preparation for rmANOVA  
  
longdat.RT <- melt(dat\_all,  
# ID variables - all the variables to keep but not split apart on  
 id.vars=c("sub\_id", "group","order"),  
# The source columns  
 measure.vars=c("p\_RT\_pre\_ST","p\_RT\_rev\_ST","p\_RT\_post\_ST","p\_RT\_pre\_CT","p\_RT\_rev\_CT","p\_RT\_post\_CT"),  
# Name of the destination column that will identify the original  
# column that the measurement came from  
 variable.name="phase",  
 value.name="p\_RT"  
)  
  
# use package strex to extract last number from sub\_id and turn it into factor  
longdat.RT$id <- str\_last\_number(longdat.RT$sub\_id)  
longdat.RT$id <- as.factor(longdat.RT$id)  
longdat.RT$p\_RT <- as.numeric(longdat.RT$p\_RT)  
  
longdat.RT <- longdat.RT %>% arrange(id)  
  
longdat.RT <- longdat.RT %>% arrange(phase)  
  
longdat.RT$volat <- gl(6,56,labels=c("pre\_ST","rev\_ST","post\_ST","pre\_CT","rev\_CT","post\_CT"))  
  
levels(longdat.RT$volat)[levels(longdat.RT$volat)=="pre\_ST"] <- "pre\_CT"  
levels(longdat.RT$volat)[levels(longdat.RT$volat)=="rev\_ST"] <- "rev\_CT"  
levels(longdat.RT$volat)[levels(longdat.RT$volat)=="post\_ST"] <- "post\_CT"  
  
longdat.RT <- longdat.RT %>% arrange(id)

## rmANOVA p\_correct

# make a boxplot to take a look at distribution of p\_correct group and phasewise  
ggplot(longdat.correct, aes(x=phase, y=p\_correct)) + geom\_boxplot()



# extract string ST or CT from phase strings and turn it into logical, then numeric  
longdat.correct <- longdat.correct %>% mutate(cond = grepl("\*CT",longdat.correct$phase))  
longdat.correct <- longdat.correct %>% mutate(cond2 = longdat.correct$cond\*1) %>% rename(cond = longdat.correct$cond2)  
  
# now ezANOVA can be filled with dv = p\_correct, within-variables = phase and cond, between-variable = group and type 3 sums of squares  
  
res.correct <- ezANOVA(longdat.correct, p\_correct, id, within = .(volat,cond), between = group, detailed = TRUE, type =2)

## Warning: Converting "cond" to factor for ANOVA.

## Warning: "group" will be treated as numeric.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a  
## well-considered value for the type argument to ezANOVA().

res.correct

## $ANOVA  
## Effect DFn DFd SSn SSd F p  
## 1 (Intercept) 1 54 1.743189e+02 2.8961323 3250.2727475 6.262742e-50  
## 2 group 1 54 3.008109e-01 2.8961323 5.6087863 2.147964e-02  
## 3 volat 2 108 1.279715e+00 2.0109938 34.3634032 2.821965e-12  
## 5 cond 1 54 1.028596e-02 0.5206182 1.0668889 3.062532e-01  
## 4 group:volat 2 108 4.340665e-02 2.0109938 1.1655726 3.156342e-01  
## 6 group:cond 1 54 5.690762e-03 0.5206182 0.5902619 4.456648e-01  
## 7 volat:cond 2 108 3.919871e-02 1.1307778 1.8719240 1.587842e-01  
## 8 group:volat:cond 2 108 5.640453e-02 1.1307778 2.6935836 7.218250e-02  
## p<.05 ges  
## 1 \* 0.9637405128  
## 2 \* 0.0438542454  
## 3 \* 0.1632656252  
## 5 0.0015658791  
## 4 0.0065748441  
## 6 0.0008669374  
## 7 0.0059412497  
## 8 0.0085268565  
##   
## $`Mauchly's Test for Sphericity`  
## Effect W p p<.05  
## 3 volat 0.7810793 0.001433575 \*  
## 4 group:volat 0.7810793 0.001433575 \*  
## 7 volat:cond 0.8054456 0.003235641 \*  
## 8 group:volat:cond 0.8054456 0.003235641 \*  
##   
## $`Sphericity Corrections`  
## Effect GGe p[GG] p[GG]<.05 HFe p[HF]  
## 3 volat 0.8203979 1.745007e-10 \* 0.8426768 1.045507e-10  
## 4 group:volat 0.8203979 3.088041e-01 0.8426768 3.098201e-01  
## 7 volat:cond 0.8371323 1.659861e-01 0.8608054 1.649697e-01  
## 8 group:volat:cond 0.8371323 8.239331e-02 0.8608054 8.083508e-02  
## p[HF]<.05  
## 3 \*  
## 4   
## 7   
## 8

ezPlot(  
 data = longdat.correct  
 , dv = .(p\_correct)  
 , wid = .(id)  
 , within = .(volat,cond)  
 , between = .(group)  
 , x = .(volat)  
)

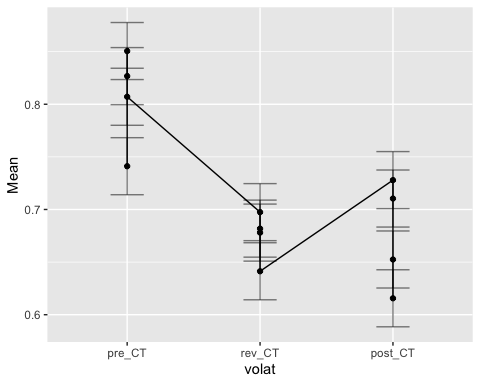
## Warning: Converting "cond" to factor for ANOVA.

## Warning: "group" will be treated as numeric.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a  
## well-considered value for the type argument to ezANOVA().

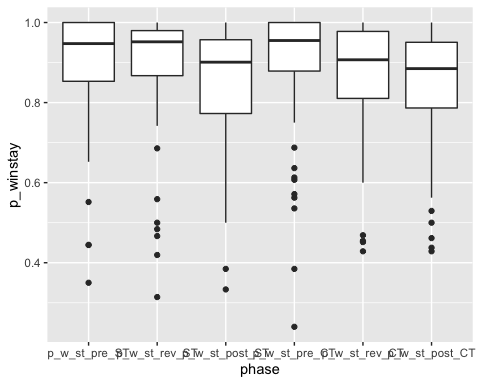
## Warning: Mixed within-and-between-Ss effect requested; FLSD is only appropriate  
## for within-Ss comparisons (see warning in ?ezStats or ?ezPlot).

## Warning in ezStats(data = data, dv = dv, wid = wid, within = within, within\_full  
## = within\_full, : Unbalanced groups. Mean N will be used in computation of FLSD



## rmANOVA p\_winstay

# make a boxplot to take a look at distribution of p\_correct group and phasewise  
ggplot(longdat.winstay, aes(x=phase, y=p\_winstay)) + geom\_boxplot()



# extract string ST or CT from phase strings and turn it into logical, then numeric  
longdat.winstay <- longdat.winstay %>% mutate(cond = grepl("\*CT",longdat.winstay$phase))  
longdat.winstay <- longdat.winstay %>% mutate(cond2 = longdat.winstay$cond\*1) %>% rename(cond = longdat.winstay$cond2)  
  
# now ezANOVA can be filled with dv = p\_correct, within-variables = phase and cond, between-variable = group and type 3 sums of squares  
  
res.winstay <- ezANOVA(longdat.winstay, p\_winstay, id, within = .(volat,cond), between = group, detailed = TRUE, type = 2)

## Warning: Converting "cond" to factor for ANOVA.

## Warning: "group" will be treated as numeric.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a  
## well-considered value for the type argument to ezANOVA().

res.winstay

## $ANOVA  
## Effect DFn DFd SSn SSd F p  
## 1 (Intercept) 1 54 2.520135e+02 5.9673272 2.280540e+03 7.444340e-46  
## 2 group 1 54 1.375040e-01 5.9673272 1.244312e+00 2.695812e-01  
## 3 volat 2 108 1.271851e-01 0.8188221 8.387651e+00 4.110899e-04  
## 5 cond 1 54 1.079049e-02 0.6968840 8.361310e-01 3.645707e-01  
## 4 group:volat 2 108 1.090940e-02 0.8188221 7.194573e-01 4.893356e-01  
## 6 group:cond 1 54 8.688749e-04 0.6968840 6.732719e-02 7.962556e-01  
## 7 volat:cond 2 108 4.292300e-03 0.6121584 3.786343e-01 6.857014e-01  
## 8 group:volat:cond 2 108 3.719712e-02 0.6121584 3.281250e+00 4.136079e-02  
## p<.05 ges  
## 1 \* 0.9688776584  
## 2 0.0167021901  
## 3 \* 0.0154681640  
## 5 0.0013311758  
## 4 0.0013458256  
## 6 0.0001073207  
## 7 0.0005299473  
## 8 \* 0.0045739479  
##   
## $`Mauchly's Test for Sphericity`  
## Effect W p p<.05  
## 3 volat 0.8879876 0.04293216 \*  
## 4 group:volat 0.8879876 0.04293216 \*  
## 7 volat:cond 0.9434452 0.21379232   
## 8 group:volat:cond 0.9434452 0.21379232   
##   
## $`Sphericity Corrections`  
## Effect GGe p[GG] p[GG]<.05 HFe p[HF]  
## 3 volat 0.8992705 0.00068928 \* 0.9283242 0.0005937075  
## 4 group:volat 0.8992705 0.47568027 0.9283242 0.4797692927  
## 7 volat:cond 0.9464724 0.67422413 0.9798286 0.6814538966  
## 8 group:volat:cond 0.9464724 0.04417393 \* 0.9798286 0.0423992770  
## p[HF]<.05  
## 3 \*  
## 4   
## 7   
## 8 \*

ezPlot(  
 data = longdat.winstay  
 , dv = .(p\_winstay)  
 , wid = .(id)  
 , within = .(volat,cond)  
 , between = .(group)  
 , x = .(volat)  
)

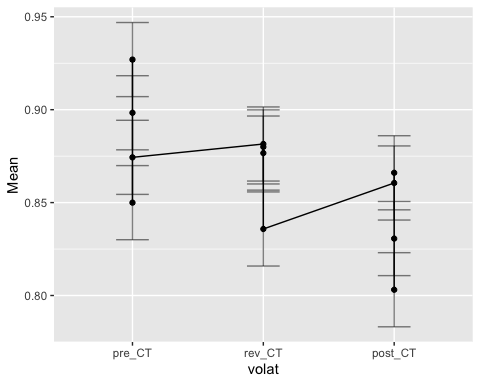
## Warning: Converting "cond" to factor for ANOVA.

## Warning: "group" will be treated as numeric.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a  
## well-considered value for the type argument to ezANOVA().

## Warning: Mixed within-and-between-Ss effect requested; FLSD is only appropriate  
## for within-Ss comparisons (see warning in ?ezStats or ?ezPlot).

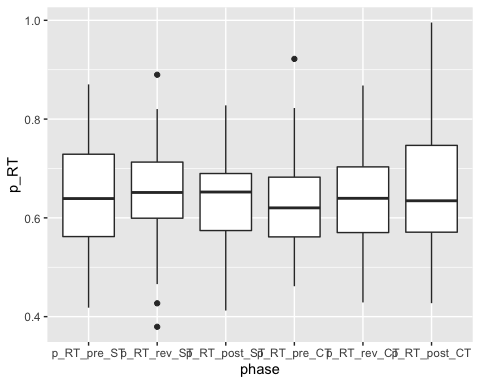
## Warning in ezStats(data = data, dv = dv, wid = wid, within = within, within\_full  
## = within\_full, : Unbalanced groups. Mean N will be used in computation of FLSD



# interaction separate for AD

## rmANOVA p\_RT

# make a boxplot to take a look at distribution of p\_correct group and phasewise  
ggplot(longdat.RT, aes(x=phase, y=p\_RT)) + geom\_boxplot()



# extract string ST or CT from phase strings and turn it into logical, then numeric  
longdat.RT <- longdat.RT %>% mutate(cond = grepl("\*CT",longdat.RT$phase))  
longdat.RT <- longdat.RT %>% mutate(cond2 = longdat.RT$cond\*1) %>% rename(cond = longdat.RT$cond2)  
  
# now ezANOVA can be filled with dv = p\_correct, within-variables = phase and cond, between-variable = group and type 3 sums of squares  
  
res.RT <- ezANOVA(longdat.RT, p\_RT, id, within = .(volat,cond), between = group, detailed = TRUE, type =2)

## Warning: Converting "cond" to factor for ANOVA.

## Warning: "group" will be treated as numeric.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a  
## well-considered value for the type argument to ezANOVA().

res.RT

## $ANOVA  
## Effect DFn DFd SSn SSd F p  
## 1 (Intercept) 1 54 1.390672e+02 2.5491039 2.945989e+03 8.510148e-49  
## 2 group 1 54 2.404000e-03 2.5491039 5.092613e-02 8.223114e-01  
## 3 volat 2 108 1.083611e-02 0.2952648 1.981781e+00 1.428041e-01  
## 5 cond 1 54 5.795893e-07 0.6458352 4.846100e-05 9.944713e-01  
## 4 group:volat 2 108 2.976976e-03 0.2952648 5.444493e-01 5.817451e-01  
## 6 group:cond 1 54 2.280202e-04 0.6458352 1.906537e-02 8.906928e-01  
## 7 volat:cond 2 108 1.560543e-02 0.1939808 4.344210e+00 1.532441e-02  
## 8 group:volat:cond 2 108 6.002621e-03 0.1939808 1.670998e+00 1.928843e-01  
## p<.05 ges  
## 1 \* 9.741916e-01  
## 2 6.520934e-04  
## 3 2.932625e-03  
## 5 1.573182e-07  
## 4 8.073896e-04  
## 6 6.188780e-05  
## 7 \* 4.217924e-03  
## 8 1.626644e-03  
##   
## $`Mauchly's Test for Sphericity`  
## Effect W p p<.05  
## 3 volat 0.5851135 6.789902e-07 \*  
## 4 group:volat 0.5851135 6.789902e-07 \*  
## 7 volat:cond 0.8260936 6.328120e-03 \*  
## 8 group:volat:cond 0.8260936 6.328120e-03 \*  
##   
## $`Sphericity Corrections`  
## Effect GGe p[GG] p[GG]<.05 HFe p[HF]  
## 3 volat 0.7067705 0.15716267 0.7201926 0.15654922  
## 4 group:volat 0.7067705 0.52210836 0.7201926 0.52531380  
## 7 volat:cond 0.8518566 0.02065372 \* 0.8767757 0.01964111  
## 8 group:volat:cond 0.8518566 0.19731612 0.8767757 0.19662665  
## p[HF]<.05  
## 3   
## 4   
## 7 \*  
## 8

ezPlot(  
 data = longdat.correct  
 , dv = .(p\_correct)  
 , wid = .(id)  
 , within = .(volat,cond)  
 , between = .(group)  
 , x = .(volat)  
)

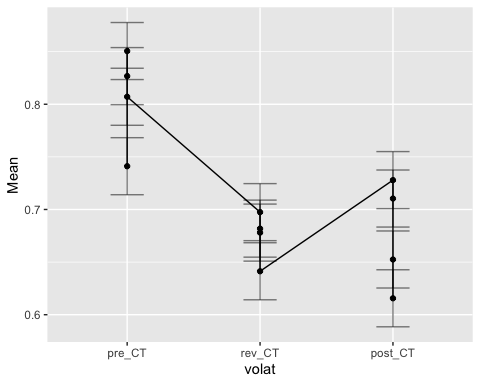
## Warning: Converting "cond" to factor for ANOVA.

## Warning: "group" will be treated as numeric.

## Warning: Data is unbalanced (unequal N per group). Make sure you specified a  
## well-considered value for the type argument to ezANOVA().

## Warning: Mixed within-and-between-Ss effect requested; FLSD is only appropriate  
## for within-Ss comparisons (see warning in ?ezStats or ?ezPlot).

## Warning in ezStats(data = data, dv = dv, wid = wid, within = within, within\_full  
## = within\_full, : Unbalanced groups. Mean N will be used in computation of FLSD

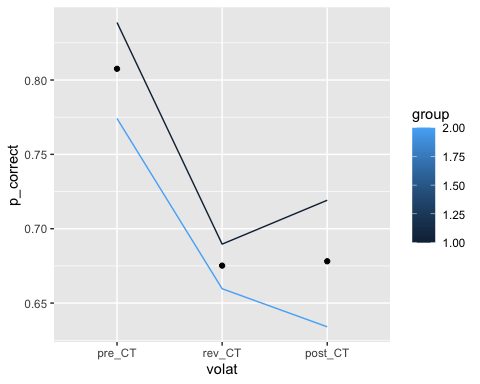


## Mixed Effects Model p\_correct

# predict p\_correct only from intercept, 2 repeated measure factors (volat and cond2) are nested within participant (id)  
basmod <- lme(p\_correct ~ 1,data = longdat.correct, random = ~1|id/volat/cond2, method = "ML")  
  
#   
mod <- lme(p\_correct ~ volat, random = ~1|id/volat/cond2,data = longdat.correct, method = "ML")  
  
# main effects  
groupmod <- update(basmod, .~. + group)  
volatmod <- update(groupmod, .~. + volat)  
condmod <- update(volatmod, .~. + cond2)  
  
# 2-way interactions  
group\_volat <- update(condmod, .~. + group:volat)  
group\_cond <- update(group\_volat, .~. + group:cond2)  
volat\_cond <- update(group\_cond, .~. + volat:cond2)  
  
# 3-way interactions  
group\_volat\_cond <- update(volat\_cond, .~. + group:volat:cond2)  
  
# compare models with anova  
anova(basmod, mod, groupmod, volatmod, condmod, group\_volat, group\_cond, volat\_cond, group\_volat\_cond)

## Model df AIC BIC logLik Test L.Ratio p-value  
## basmod 1 5 -355.9801 -336.8946 182.9901   
## mod 2 7 -406.2131 -379.4934 210.1066 1 vs 2 54.23300 <.0001  
## groupmod 3 6 -359.5140 -336.6113 185.7570 2 vs 3 48.69914 <.0001  
## volatmod 4 8 -409.7470 -379.2101 212.8735 3 vs 4 54.23300 <.0001  
## condmod 5 9 -408.7300 -374.3760 213.3650 4 vs 5 0.98305 0.3214  
## group\_volat 6 11 -407.1218 -365.1336 214.5609 5 vs 6 2.39176 0.3024  
## group\_cond 7 12 -405.6682 -359.8628 214.8341 6 vs 7 0.54636 0.4598  
## volat\_cond 8 14 -405.4806 -352.0411 216.7403 7 vs 8 3.81247 0.1486  
## group\_volat\_cond 9 16 -407.1230 -346.0492 219.5615 8 vs 9 5.64233 0.0595

ggplot(longdat.correct, aes(x = volat,y = p\_correct, col = group)) + stat\_summary(fun = mean, geom = "point") + stat\_summary(fun = mean, geom = "line", aes(group = group))

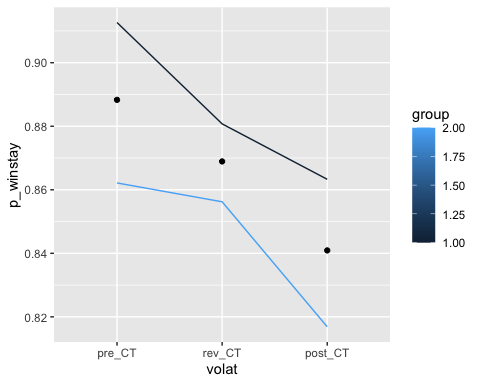


## Mixed Effects Model p\_winstay

# predict p\_winstay only from intercept, 2 repeated measure factors (volat and cond2) are nested within participant (id)  
basmod <- lme(p\_winstay ~ 1,data = longdat.winstay, random = ~1|id/volat/cond2, method = "ML")  
  
#   
mod <- lme(p\_winstay ~ volat, random = ~1|id/volat/cond2,data = longdat.winstay, method = "ML")  
  
# main effects  
groupmod <- update(basmod, .~. + group)  
volatmod <- update(groupmod, .~. + volat)  
condmod <- update(volatmod, .~. + cond2)  
  
# 2-way interactions  
group\_volat <- update(condmod, .~. + group:volat)  
group\_cond <- update(group\_volat, .~. + group:cond2)  
volat\_cond <- update(group\_cond, .~. + volat:cond2)  
  
# 3-way interactions  
group\_volat\_cond <- update(volat\_cond, .~. + group:volat:cond2)  
  
# compare models with anova  
anova(basmod, mod, groupmod, volatmod, condmod, group\_volat, group\_cond, volat\_cond, group\_volat\_cond)

## Model df AIC BIC logLik Test L.Ratio p-value  
## basmod 1 5 -502.8868 -483.8012 256.4434   
## mod 2 7 -514.5874 -487.8677 264.2937 1 vs 2 15.700632 0.0004  
## groupmod 3 6 -502.1626 -479.2599 257.0813 2 vs 3 14.424876 0.0001  
## volatmod 4 8 -513.8632 -483.3263 264.9316 3 vs 4 15.700632 0.0004  
## condmod 5 9 -513.2450 -478.8910 265.6225 4 vs 5 1.381800 0.2398  
## group\_volat 6 11 -510.6490 -468.6608 266.3245 5 vs 6 1.403994 0.4956  
## group\_cond 7 12 -508.7611 -462.9558 266.3806 6 vs 7 0.112124 0.7377  
## volat\_cond 8 14 -505.3157 -451.8761 266.6578 7 vs 8 0.554559 0.7578  
## group\_volat\_cond 9 16 -506.1680 -445.0943 269.0840 8 vs 9 4.852380 0.0884

ggplot(longdat.winstay, aes(x = volat,y = p\_winstay, col = group)) + stat\_summary(fun = mean, geom = "point") + stat\_summary(fun = mean, geom = "line", aes(group = group))

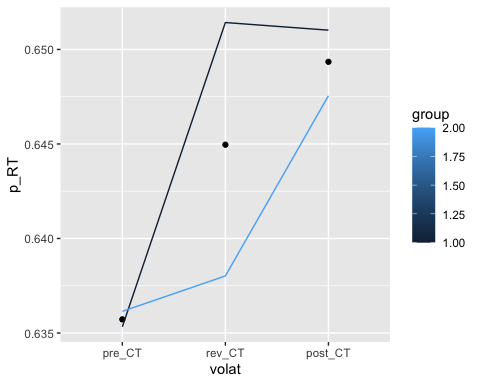


## Mixed Effects Model p\_RT

# predict p\_RT only from intercept, 2 repeated measure factors (volat and cond2) are nested within participant (id)  
basmod <- lme(p\_RT ~ 1,data = longdat.RT, random = ~1|id/volat/cond2, method = "ML")  
  
#   
mod <- lme(p\_RT ~ volat, random = ~1|id/volat/cond2,data = longdat.RT, method = "ML")  
  
# main effects  
groupmod <- update(basmod, .~. + group)  
volatmod <- update(groupmod, .~. + volat)  
condmod <- update(volatmod, .~. + cond2)  
  
# 2-way interactions  
group\_volat <- update(condmod, .~. + group:volat)  
group\_cond <- update(group\_volat, .~. + group:cond2)  
volat\_cond <- update(group\_cond, .~. + volat:cond2)  
  
# 3-way interactions  
group\_volat\_cond <- update(volat\_cond, .~. + group:volat:cond2)  
  
# compare models with anova  
anova(basmod, mod, groupmod, volatmod, condmod, group\_volat, group\_cond, volat\_cond, group\_volat\_cond)

## Model df AIC BIC logLik Test L.Ratio p-value  
## basmod 1 5 -743.0439 -723.9584 376.5220   
## mod 2 7 -741.6476 -714.9279 377.8238 1 vs 2 2.603707 0.2720  
## groupmod 3 6 -741.0967 -718.1940 376.5484 2 vs 3 2.550920 0.1102  
## volatmod 4 8 -739.7004 -709.1635 377.8502 3 vs 4 2.603707 0.2720  
## condmod 5 9 -737.7006 -703.3466 377.8503 4 vs 5 0.000140 0.9906  
## group\_volat 6 11 -734.4201 -692.4319 378.2101 5 vs 6 0.719570 0.6978  
## group\_cond 7 12 -732.4753 -686.6700 378.2377 6 vs 7 0.055191 0.8143  
## volat\_cond 8 14 -732.2786 -678.8391 380.1393 7 vs 8 3.803326 0.1493  
## group\_volat\_cond 9 16 -729.7555 -668.6817 380.8777 8 vs 9 1.476816 0.4779

ggplot(longdat.RT, aes(x = volat,y = p\_RT, col = group)) + stat\_summary(fun = mean, geom = "point") + stat\_summary(fun = mean, geom = "line", aes(group = group))



## Plots for Descriptive Stats

```