Experiment 3

General preparations

Load libraries

```
library(mousetrap)
library(ggplot2)
library(dplyr)
library(tidyr)
library(afex)
library(MBESS)
library(ordinal)
```

Custom ggplot2 theme

```
theme_set(theme_classic()+
  theme(
    axis.line = element_line(colour = "black"),
    axis.ticks = element_line(colour = "black"),
    axis.text = element_text(colour = "black"),
    panel.border = element_rect(colour = "black", fill=NA)
  ))
```

Custom functions

```
# Function to compute confidence interval for partial eta-squared
get_partial_etas <- function(anova_table, conf.level=.90){
  partial_etas <- sapply(row.names(anova_table),function(i){
    F <- anova_table[i,"F"]
    df1 <- anova_table[i,"num Df"]
    df2 <- anova_table[i,"den Df"]
    ci <- conf.limits.ncf(F.value=F,conf.level=conf.level,df.1=df1,df.2=df2)
    return(
      c(pes=((F*df1)/(F*df1+df2)),
        lower=ci$Lower.Limit/(ci$Lower.Limit+df1+df2+1),
        upper=ci$Upper.Limit/(ci$Upper.Limit+df1+df2+1)))
  })
  return(t(partial_etas))
}
```

Data import

```
raw_data <- read.csv("../data/exp3.csv")
raw_data$Typicality <- factor(raw_data$Condition,levels=c("Typical","Atypical"))
raw_data$group <- factor(raw_data$group,levels=c("static","rtmax","initmax","dynamic"))
```

Correctness - analysis including all trials

Percent of correct trials per condition

```
with(raw_data,table(group, correct)/c(table(group)))
```

```
##          correct
## group              0          1
##   static  0.05887600 0.94112400
##   rtmax   0.10877193 0.89122807
##   initmax 0.10287081 0.89712919
##   dynamic 0.06403509 0.93596491
```

Chi-squared test

```
chisq.test(with(raw_data,table(group, correct)),correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
##
## data:  with(raw_data, table(group, correct))
## X-squared = 29.927, df = 3, p-value = 1.43e-06
```

Generalized linear mixed model

```
# use default contrasts (dummy coding with static as baseline)
contrasts(raw_data$group)
```

```
##         rtmax initmax dynamic
## static      0       0       0
## rtmax       1       0       0
## initmax     0       1       0
## dynamic     0       0       1
```

```
summary(glmer(correct~(1|subject_nr)+group,family="binomial",data=raw_data))
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: correct ~ (1 | subject_nr) + group
##    Data: raw_data
##
##      AIC      BIC   logLik deviance df.resid
##   2642.0   2674.2  -1316.0   2632.0     4650
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3273  0.2066  0.2595  0.3245  0.5809
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  subject_nr (Intercept) 0.386    0.6213
```

```
## Number of obs: 4655, groups:  subject_nr, 245
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   2.94406    0.16029  18.367  < 2e-16 ***
## grouprtmax   -0.70835    0.20090  -3.526 0.000422 ***
## groupinitmax -0.61969    0.19833  -3.125 0.001781 **
## groupdynamic -0.09055    0.21482  -0.422 0.673380
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) grprtm grpntm
## grouprtmax  -0.761
## groupinitmx -0.760  0.602
## groupdynamc -0.694  0.553  0.560
```

Correctness - analysis excluding trials in rtmax condition > limit

Exclude trials in rtmax condition above time limit

```
# Count eligible and non-eligible trials in rtmax condition
n_eligible <- sum(with(raw_data,group=="rtmax" & response!="None"))
n_noneligible <- sum(with(raw_data,group=="rtmax" & response=="None"))

# Percent trials in rtmax condition above total time limit
n_noneligible/(n_eligible+n_noneligible)
```

```
## [1] 0.0377193
```

```
# Exclude non-eligible trials
raw_data <- subset(raw_data, response!="None")
```

Percent of correct trials per condition

```
with(raw_data,table(group, correct)/c(table(group)))
```

```
##          correct
## group              0          1
##   static  0.05887600 0.94112400
##   rtmax   0.07383774 0.92616226
##   initmax 0.10287081 0.89712919
##   dynamic 0.06403509 0.93596491
```

Chi-squared test

```
chisq.test(with(raw_data,table(group, correct)),correct = FALSE)
```

```
##
##  Pearson's Chi-squared test
```

```
##
## data:  with(raw_data, table(group, correct))
## X-squared = 20.044, df = 3, p-value = 0.0001662
```

Generalized linear mixed model

```
summary(glmer(correct~(1|subject_nr)+group,family="binomial",data=raw_data))
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: correct ~ (1 | subject_nr) + group
##    Data: raw_data
##
##      AIC      BIC   logLik deviance df.resid
##   2430.3   2462.5  -1210.2   2420.3     4607
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.3768  0.1987  0.2359  0.2906  0.6089
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  subject_nr (Intercept) 0.4761   0.69
## Number of obs: 4612, groups:  subject_nr, 245
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.9800     0.1676  17.785  < 2e-16 ***
## grouprtmax    -0.2606     0.2204  -1.182  0.23706
## groupinitmax  -0.6258     0.2073  -3.019  0.00254 **
## groupdynamic  -0.0915     0.2237  -0.409  0.68248
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) grprtm grpntm
## grouprtmax  -0.707
## groupinitmx -0.756  0.567
## groupdynamc -0.691  0.524  0.557
```

Exclude incorrect trials

```
raw_data <- subset(raw_data, correct==1)
```

Trajectory preprocessing

```
mt_data <- mt_import_mousetrap(raw_data,
  xpos_label = c("xpos_initial_phase","xpos_get_response"),
```

```
  ypos_label = c("ypos_initial_phase","ypos_get_response"),
  timestamps_label = c("timestamps_initial_phase","timestamps_get_response"))
mt_data <- mt_remap_symmetric(mt_data)
mt_data <- mt_align_start(mt_data, start=c(0,0))
mt_data <- mt_derivatives(mt_data)
mt_data <- mt_measures(mt_data)
mt_data <- mt_time_normalize(mt_data)
```

Manipulation check using per participant mean of time variables

Aggregate data per participant and condition

```
mt_data$measures$RT_initial <- mt_data$data$response_time_initial_phase
mt_data$measures$IT <- mt_data$measures$initiation_time
mt_data$measures$RT_post <- mt_data$data$response_time_get_response

agg_times <- mt_aggregate_per_subject(mt_data,
  use_variables = c("RT_initial","IT","RT","RT_post"),
  use2_variables = "group",subject_id="subject_nr")
```

Descriptives

```
mean_times <- agg_times %>%
  group_by(group) %>%
  summarize(
    N = n(),
    M_RT_inital = mean(RT_initial),
    SD_RT_initial = sd(RT_initial),
    M_IT = mean(IT),
    SD_IT = sd(IT),
    M_RT = mean(RT),
    SD_RT = sd(RT)
    ) %>%
  as.data.frame()
print(mean_times, digits=5)
```

```
##     group  N M_RT_inital SD_RT_initial   M_IT  SD_IT   M_RT   SD_RT
## 1  static 59      808.47        324.06 508.69 215.60 2110.4  654.06
## 2   rtmax 60      650.12        176.61 437.25 159.98 1521.6  183.42
## 3 initmax 66      377.43        159.49 243.14 142.76 1471.7  248.62
## 4 dynamic 60      773.38        752.17 348.67 233.25 2805.4 1199.82
```

```
# RT post (interesting mostly for dynamic condition)
agg_times %>%
  group_by(group) %>%
  summarize(
    M_RT_post = mean(RT_post),
    SD_RT_post = sd(RT_post)
  )%>%
  as.data.frame()
```

```
##     group M_RT_post SD_RT_post
## 1  static 1291.0419   584.8519
## 2   rtmax  860.5363   190.2649
## 3 initmax 1083.3367   308.9183
## 4 dynamic 2021.0729   671.1218
```

Specify contrasts (used in contrast analyses later)

```
contrast_matrix_separate <- list(
  rtmax_vs_static = c(-1,1,0,0),
  initmax_vs_static = c(-1,0,1,0),
  dynamic_vs_static= c(-1,0,0,1))
```

Compare RT initial

```
# ANOVA
anova_RT_initial <- aov_ez(data=agg_times,dv = "RT_initial", between = "group", id = "subject_nr")
```

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

```
nice(anova_RT_initial,es = c("pes","ges"))
```

```
## Anova Table (Type 3 tests)
##
## Response: RT_initial
##   Effect     df       MSE         F ges pes p.value
## 1  group 3, 241 178274.40 13.64 *** .15 .15  <.0001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
# Partial eta-squared including 90 % CI
round(get_partial_etas(anova_RT_initial$anova_table, conf.level=.90),2)
```

```
##        pes lower upper
## group 0.15  0.08  0.21
```

```
# Contrast analysis
anova_RT_initial_grid <- lsmeans(anova_RT_initial,~group)
contrast(anova_RT_initial_grid,contrast_matrix_separate)
```

```
##  contrast            estimate       SE  df t.ratio p.value
##  rtmax_vs_static   -158.35691 77.41344 241  -2.046  0.0419
##  initmax_vs_static -431.04524 75.64871 241  -5.698  <.0001
##  dynamic_vs_static  -35.09692 77.41344 241  -0.453  0.6507
```

Compare initiation time

```
# ANOVA
anova_IT <- aov_ez(data=agg_times,dv = "IT", between = "group", id = "subject_nr")
```

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

```r
nice(anova_IT,es = c("pes","ges"))
```

```
## Anova Table (Type 3 tests)
##
## Response: IT
##   Effect     df      MSE         F ges pes p.value
## 1  group 3, 241 36268.60 22.69 *** .22 .22  <.0001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```r
# Partial eta-squared including 90 % CI
round(get_partial_etas(anova_IT$anova_table, conf.level=.90),2)
```

```
##        pes lower upper
## group 0.22  0.14  0.29
```

```r
# Contrast analysis
anova_IT_grid <- lsmeans(anova_IT,~group)
contrast(anova_IT_grid,contrast_matrix_separate)
```

```
##  contrast           estimate       SE  df t.ratio p.value
##  rtmax_vs_static    -71.4404 34.91702 241  -2.046  0.0418
##  initmax_vs_static -265.5558 34.12105 241  -7.783  <.0001
##  dynamic_vs_static -160.0251 34.91702 241  -4.583  <.0001
```

Compare total RT

```r
# ANOVA
anova_RT <- aov_ez(data=agg_times,dv = "RT", between = "group", id = "subject_nr")
```

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

```r
nice(anova_RT,es = c("pes","ges"))
```

```
## Anova Table (Type 3 tests)
##
## Response: RT
##   Effect     df       MSE         F ges pes p.value
## 1  group 3, 241 480287.20 49.61 *** .38 .38  <.0001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```r
# Partial eta-squared including 90 % CI
round(get_partial_etas(anova_RT$anova_table, conf.level=.90),2)
```

```
##        pes lower upper
## group 0.38   0.3  0.44
```

```r
# Contrast analysis
anova_RT_grid <- lsmeans(anova_RT,~group)
contrast(anova_RT_grid,contrast_matrix_separate)
```

```
##  contrast           estimate       SE  df t.ratio p.value
##  rtmax_vs_static   -588.8280 127.0640 241  -4.634  <.0001
##  initmax_vs_static -638.7678 124.1674 241  -5.144  <.0001
##  dynamic_vs_static  694.9442 127.0640 241   5.469  <.0001
```

Manipulation check using per participant median of time variables

Aggregate data per participant and condition

```
agg_times <- mt_aggregate_per_subject(mt_data,
  use_variables = c("IT","RT_initial","RT"),
  use2_variables = "group",subject_id="subject_nr",
  .funs="median")
```

Descriptives

```
mean_times <- agg_times %>%
  group_by(group) %>%
  summarize(
    N = n(),
    M_RT_inital = mean(RT_initial),
    SD_RT_initial = sd(RT_initial),
    M_IT = mean(IT),
    SD_IT = sd(IT),
    M_RT = mean(RT),
    SD_RT = sd(RT)
    ) %>%
  as.data.frame()

print(mean_times, digits=5)
```

```
##     group  N M_RT_inital SD_RT_initial   M_IT  SD_IT   M_RT  SD_RT
## 1  static 59      760.84        297.31 497.36 207.87 1934.0 588.13
## 2   rtmax 60      630.30        178.20 437.12 163.29 1476.9 206.95
## 3 initmax 66      355.22        166.27 231.51 152.07 1376.4 238.00
## 4 dynamic 60      516.82        329.58 267.10 164.41 2461.7 859.45
```

Compare RT initial

```
# ANOVA
anova_RT_initial <- aov_ez(data=agg_times,dv = "RT_initial", between = "group", id = "subject_nr")
```

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

```
nice(anova_RT_initial,es = c("pes","ges"))
```

```
## Anova Table (Type 3 tests)
##
## Response: RT_initial
##   Effect     df      MSE         F ges pes p.value
## 1  group 3, 241 63095.60 29.35 *** .27 .27  <.0001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
# Partial eta-squared including 90 % CI
round(get_partial_etas(anova_RT_initial$anova_table, conf.level=.90),2)
```

```
##        pes lower upper
## group 0.27  0.19  0.33
```

```
# Contrast analysis
anova_RT_initial_grid <- lsmeans(anova_RT_initial,~group)
contrast(anova_RT_initial_grid,contrast_matrix_separate)
```

```
##  contrast           estimate       SE  df t.ratio p.value
##  rtmax_vs_static   -130.5390 46.05443 241  -2.834  0.0050
##  initmax_vs_static -405.6193 45.00456 241  -9.013  <.0001
##  dynamic_vs_static -244.0223 46.05443 241  -5.299  <.0001
```

Compare initiation time

```
# ANOVA
anova_IT <- aov_ez(data=agg_times,dv = "IT", between = "group", id = "subject_nr")
```

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

```
nice(anova_IT,es = c("pes","ges"))
```

```
## Anova Table (Type 3 tests)
##
## Response: IT
##   Effect     df      MSE         F ges pes p.value
## 1  group 3, 241 29781.04 34.37 *** .30 .30  <.0001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
# Partial eta-squared including 90 % CI
round(get_partial_etas(anova_IT$anova_table, conf.level=.90),2)
```

```
##       pes lower upper
## group 0.3  0.22  0.36
```

```
# Contrast analysis
anova_IT_grid <- lsmeans(anova_IT,~group)
contrast(anova_IT_grid,contrast_matrix_separate)
```

```
##  contrast           estimate       SE  df t.ratio p.value
##  rtmax_vs_static    -60.24774 31.64039 241  -1.904  0.0581
##  initmax_vs_static -265.85683 30.91911 241  -8.598  <.0001
##  dynamic_vs_static -230.26441 31.64039 241  -7.278  <.0001
```

Compare total RT

```
# ANOVA
anova_RT <- aov_ez(data=agg_times,dv = "RT", between = "group", id = "subject_nr")
```

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

```
nice(anova_RT,es = c("pes","ges"))
```

```
## Anova Table (Type 3 tests)
##
## Response: RT
```

```
##   Effect     df       MSE         F ges pes p.value
## 1  group 3, 241 289840.97 52.25 *** .39 .39  <.0001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
# Partial eta-squared including 90 % CI
round(get_partial_etas(anova_RT$anova_table, conf.level=.90),2)
```

```
##        pes lower upper
## group 0.39  0.31  0.46
```

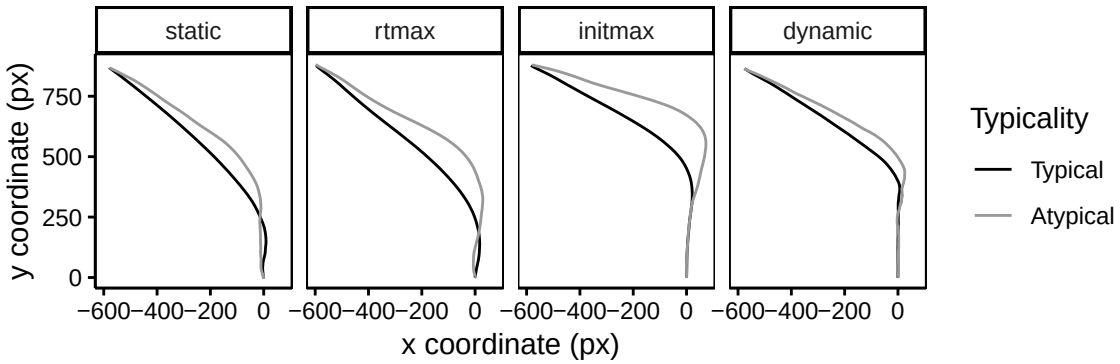
```
# Contrast analysis
anova_RT_grid <- lsmeans(anova_RT,~group)
contrast(anova_RT_grid,contrast_matrix_separate)
```

```
##  contrast           estimate       SE  df t.ratio p.value
##  rtmax_vs_static   -457.0743 98.70786 241  -4.631  <.0001
##  initmax_vs_static -557.5334 96.45770 241  -5.780  <.0001
##  dynamic_vs_static  527.6924 98.70786 241   5.346  <.0001
```

Aggregate trajectory curvature

Average time-normalized tajectories

```
mt_plot_aggregate(mt_data, use = "tn_trajectories", facet_col = "group",
  x = "xpos", y = "ypos", color = "Typicality", subject_id = "subject_nr")+
  xlab("x coordinate (px)") + ylab("y coordinate (px)")+
  scale_color_manual(values = c("black","grey60"))
```



Comparison of MAD aggregated per participant

Aggregate data per participant and condition

```
agg_mad <- mt_aggregate_per_subject(mt_data, subject_id = "subject_nr",
  use_variables = "MAD", use2_variables = c("Typicality","group"))
```

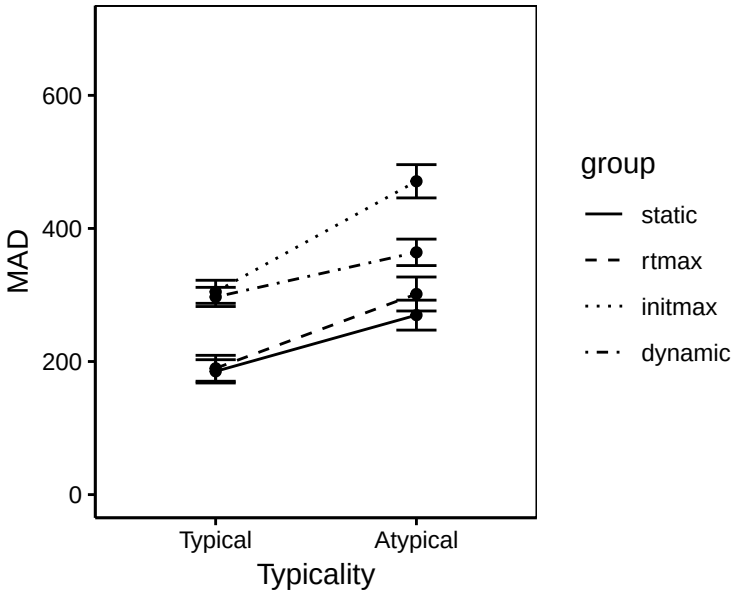
Descriptives and paired t-tests

```
mad_table <- agg_mad %>%
  group_by(group) %>%
  select(MAD,group,Typicality) %>%
  summarize(
    N =  length(MAD[Typicality=="Typical"]),
    M_t = mean(MAD[Typicality=="Typical"]),
    SD_t = sd(MAD[Typicality=="Typical"]),
    M_a = mean(MAD[Typicality=="Atypical"]),
    SD_a = sd(MAD[Typicality=="Atypical"]),
    t = t.test(MAD[Typicality=="Atypical"],MAD[Typicality=="Typical"],paired=TRUE)$statistic,
    p = t.test(MAD[Typicality=="Atypical"],MAD[Typicality=="Typical"],paired=TRUE)$p.value,
    d = (M_a-M_t)/sd(MAD[Typicality=="Atypical"]-MAD[Typicality=="Typical"])
    )

mad_table %>%
  as.data.frame() %>%
  print(digits=3)
```

```
##     group  N M_t SD_t M_a SD_a    t        p     d
## 1  static 59 185  134 270  173 4.18 1.01e-04 0.544
## 2   rtmax 60 190  151 301  198 4.32 6.00e-05 0.558
## 3 initmax 66 305  141 471  203 7.39 3.50e-10 0.910
## 4 dynamic 60 297  112 364  154 3.95 2.09e-04 0.510
```

Figure

```
ggplot(agg_mad,aes(x=Typicality,y=MAD,linetype=group,group=group))+
  geom_line(stat="summary",fun.y="mean")+
  geom_point(stat="summary",fun.y="mean")+
  geom_errorbar(stat="summary",fun.data="mean_se",width=.2,linetype=1)+
  scale_linetype_manual(values=c(1,2,3,4))+
  coord_cartesian(ylim=c(0,700))
```



ANOVA

```
anova_mad <- aov_ez(data=agg_mad, dv = "MAD", between = "group", within = "Typicality",
                    id = "subject_nr")
```

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

```
nice(anova_mad,es = c("pes","ges"))
```

```
## Anova Table (Type 3 tests)
##
## Response: MAD
##              Effect     df      MSE         F ges pes p.value
## 1             group 3, 241 37593.62 18.67 *** .14 .19  <.0001
## 2       Typicality 1, 241 14412.64 97.72 *** .10 .29  <.0001
## 3 group:Typicality 3, 241 14412.64   4.12 ** .01 .05    .007
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

```
# 90 % confidence interval for partial eta-squared
round(get_partial_etas(anova_mad$anova_table, conf.level=.90),2)
```

```
##                   pes lower upper
## group            0.19  0.11  0.25
## Typicality       0.29  0.21  0.36
## group:Typicality 0.05  0.01  0.09
```

Contrast analyses

```
# Retrieve grid
anova_mad_grid <- lsmeans(anova_mad,~Typicality:group)

# Specify contrasts
```

```r
contrast_matrix_complete <- list(
  typicality_static = c(-1,1,0,0,0,0,0,0),
  rtmax_static_main=  c(-1,-1,1,1,0,0,0,0)/2,
  initmax_static_main = c(-1,-1,0,0,1,1,0,0)/2,
  dynamic_static_main = c(-1,-1,0,0,0,0,1,1)/2,
  rtmax_static_int = c(1,-1,-1,1,0,0,0,0),
  initmax_static_int = c(1,-1,0,0,-1,1,0,0),
  dynamic_static_int = c(1,-1,0,0,0,0,-1,1))

# Test contrasts
contrast(anova_mad_grid,contrast_matrix_complete)
```

```
##  contrast             estimate       SE  df t.ratio p.value
##  typicality_static    84.48112 22.10349 241   3.822  0.0002
##  rtmax_static_main    18.18135 25.13703 241   0.723  0.4702
##  initmax_static_main 160.36694 24.56400 241   6.529  <.0001
##  dynamic_static_main 103.08993 25.13703 241   4.101  0.0001
##  rtmax_static_int     27.18582 31.12853 241   0.873  0.3833
##  initmax_static_int   81.63197 30.41892 241   2.684  0.0078
##  dynamic_static_int  -17.42019 31.12853 241  -0.560  0.5763
```

Distribution of trajectory shapes

Bimodality coefficient

```r
# Standardize MAD per participant
mt_data <- mt_standardize(mt_data, use_variables = "MAD", within = "subject_nr")

# Calculate bimodality coefficient
mt_check_bimodality(mt_data, use_variables = "z_MAD",
  grouping_variables = c("group","Typicality"), methods = "BC")
```

```
## $BC
##     group Typicality     z_MAD
## 1  static    Typical 0.5202425
## 2  static   Atypical 0.5479891
## 3   rtmax    Typical 0.5356378
## 4   rtmax   Atypical 0.5014132
## 5 initmax    Typical 0.5097199
## 6 initmax   Atypical 0.4731716
## 7 dynamic    Typical 0.5596031
## 8 dynamic   Atypical 0.5080918
```

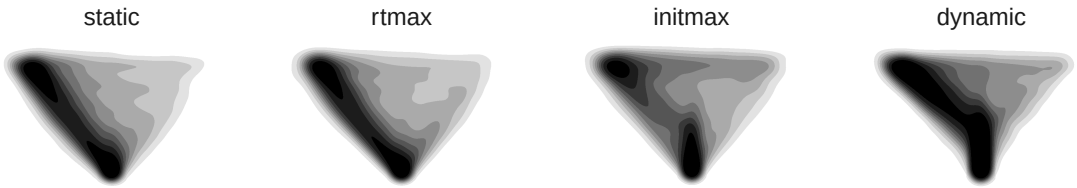
Smoothed heatmaps

```r
heatmap_smoothed <- mt_heatmap_ggplot(mt_data,
  xres = 1000,
  smooth_radius = 20,
  n_shades = 10,
  mean_image = 0.2,
```

```
  colors=c("white","black"),
  facet_col="group")
```

```
## spatializing trajectories
## calculate image
## smooth image
## enhance image by 4
## spatializing trajectories
## calculate image
## smooth image
## enhance image by 3.8
## spatializing trajectories
## calculate image
## smooth image
## enhance image by 3.7
## spatializing trajectories
## calculate image
## smooth image
## enhance image by 6.1
```

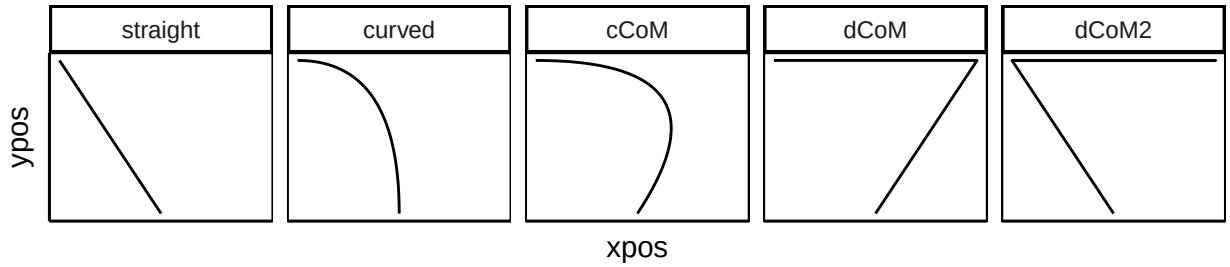
```
heatmap_smoothed+
  theme(strip.background = element_rect(colour = NA))
```



Prototype classification (standard set)

Plot prototypes

```
mt_plot(mt_prototypes,facet_col="mt_id",only_ggplot = TRUE)+
  geom_path()+
  facet_grid(cols = vars(factor(mt_id,levels=rownames(mt_prototypes))))+
  theme(axis.text=ggplot2::element_blank(),axis.ticks=ggplot2::element_blank())
```



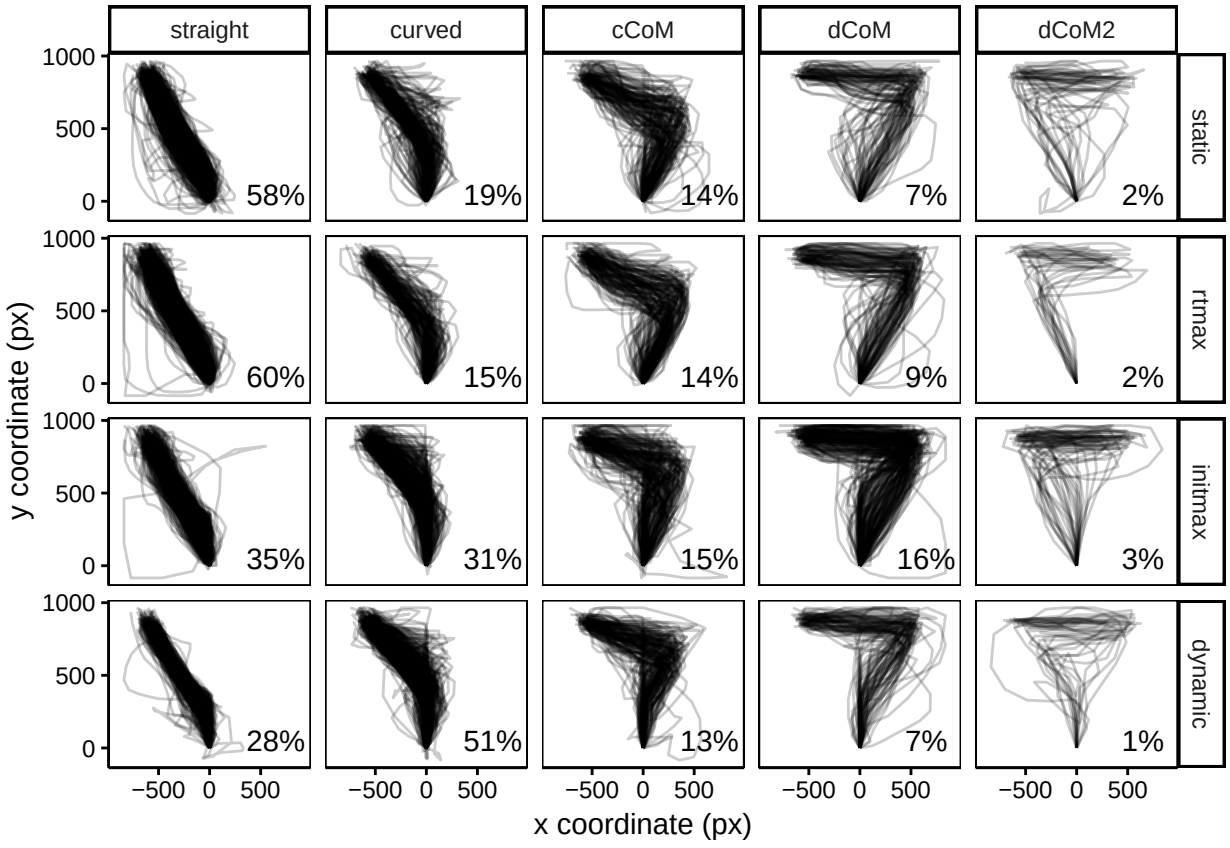
Map trajectories onto prototypes

```
mt_data <- mt_spatialize(mt_data)
mt_data <- mt_map(mt_data,prototypes = mt_prototypes,
  save_as = "measures", grouping_variables = "group")
mt_data$data$prototype_label <- mt_data$measures$prototype_label
```

Classified trajectories per group

Relative frequencies

```
prototype_percentages <- mt_data$data %>%
  group_by(group,prototype_label) %>%
  summarise(n=n()) %>%
  mutate(Percent=paste(round(100*n/sum(n)),"%",sep=""))

mt_plot(mt_data, use = "sp_trajectories",
  x = "xpos", y = "ypos", facet_col = "prototype_label", facet_row="group",alpha=.2)+
  xlab("x coordinate (px)") + ylab("y coordinate (px)")+
  geom_text(data=prototype_percentages,aes(label=Percent),x=650,y=50)+
  scale_y_continuous(breaks=c(0,500,1000))+
  coord_cartesian(xlim=c(-900,900))
```



Chi-squared test

```r
chisq.test(with(mt_data$data,table(group, prototype_label)))
```

```
##
##  Pearson's Chi-squared test
##
## data:  with(mt_data$data, table(group, prototype_label))
## X-squared = 535.73, df = 12, p-value < 2.2e-16
```

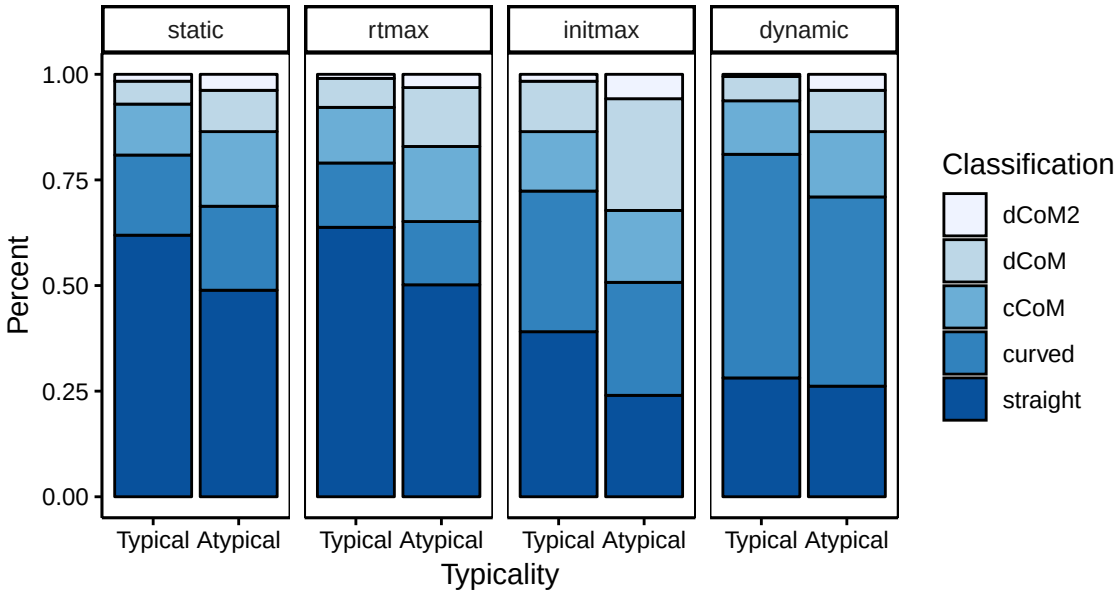
Classified trajectories per group X typicality condition

Relative frequencies

```r
rel_freq_agg <- mt_data$data %>%
  group_by(group,Typicality,prototype_label) %>%
  summarise(n=n()) %>%
  mutate(Percent=n/sum(n))

spread(rel_freq_agg[,-4],"prototype_label","Percent",fill = 0) %>%
  as.data.frame()%>%
  print(digits=2)
```

```
##     group Typicality straight curved cCoM  dCoM  dCoM2
## 1  static    Typical     0.62   0.19 0.12 0.054 0.0163
## 2  static   Atypical     0.49   0.20 0.18 0.098 0.0379
## 3   rtmax    Typical     0.64   0.15 0.13 0.069 0.0096
## 4   rtmax   Atypical     0.50   0.15 0.18 0.139 0.0314
## 5 initmax    Typical     0.39   0.33 0.14 0.119 0.0163
## 6 initmax   Atypical     0.24   0.27 0.17 0.264 0.0578
## 7 dynamic    Typical     0.28   0.53 0.13 0.057 0.0053
## 8 dynamic   Atypical     0.26   0.45 0.15 0.098 0.0379
```

```r
ggplot(rel_freq_agg,aes(x=Typicality,y=Percent,fill=forcats::fct_rev(prototype_label)))+
  geom_bar(stat="identity",color="black")+
  scale_fill_brewer(type="seq",name="Classification")+
  facet_grid(.~group)
```



Ordinal mixed regression

```
contrasts(mt_data$data$Typicality) <- c(-0.5,0.5)
# use default contrasts for group (dummy coding with static as baseline)
contrasts(mt_data$data$group)
```

```
##         rtmax initmax dynamic
## static      0       0       0
## rtmax       1       0       0
## initmax     0       1       0
## dynamic     0       0       1
```

```
summary(clmm(prototype_label~Typicality*group+(1|subject_nr),data=mt_data$data))
```

```
## Cumulative Link Mixed Model fitted with the Laplace approximation
##
## formula: prototype_label ~ Typicality * group + (1 | subject_nr)
## data:    mt_data$data
##
##  link  threshold nobs logLik   AIC      niter      max.grad cond.H
##  logit flexible  4263 -5228.98 10481.96 1398(5596) 4.29e-03 2.4e+02
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  subject_nr (Intercept) 0.6917   0.8317
## Number of groups:  subject_nr 245
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## Typicality1                0.69430    0.13713   5.063 4.13e-07 ***
## grouprtmax                 0.05609    0.18341   0.306   0.7598
## groupinitmax               1.05667    0.17656   5.985 2.17e-09 ***
## groupdynamic               0.77971    0.17836   4.371 1.23e-05 ***
## Typicality1:grouprtmax     0.15932    0.20074   0.794   0.4274
```

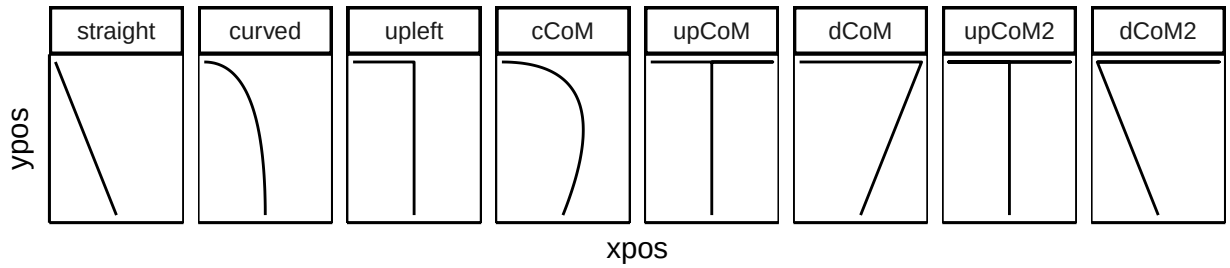
```
## Typicality1:groupinitmax  0.30876    0.18412   1.677   0.0935 .
## Typicality1:groupdynamic -0.39671    0.18112  -2.190   0.0285 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                 Estimate Std. Error z value
## straight|curved   0.1227     0.1296   0.947
## curved|cCoM       1.6248     0.1323  12.284
## cCoM|dCoM         2.6964     0.1371  19.669
## dCoM|dCoM2        4.6552     0.1692  27.506
```

Prototype classification (extended prototype set)

Extend prototype set

Include prototypes that move up all the way to the top of the screen and then... * left to the chosen option (upleft) * right to the non-chosen option and then left (upCoM) * left to the chosen option, then right to the non-chosen option, then left again (upCoM2)

```
mt_prototypes_ext <- mt_add_trajectory(mt_prototypes,
   xpos = c(0,0,-1), ypos = c(0,1.5,1.5), id = "upleft"
)

mt_prototypes_ext <- mt_add_trajectory(mt_prototypes_ext,
   xpos = c(0,0,1,-1), ypos = c(0,1.5,1.5,1.5), id = "upCoM"
)

mt_prototypes_ext <- mt_add_trajectory(mt_prototypes_ext,
   xpos = c(0,0,-1,1,-1), ypos = c(0,1.5,1.5,1.5,1.5), id = "upCoM2"
)

prototype_labels_extended <-
  c("straight","curved","upleft","cCoM","upCoM","dCoM","upCoM2","dCoM2")

mt_plot(mt_prototypes_ext,facet_col="mt_id",only_ggplot = TRUE)+
  geom_path()+
  facet_grid(cols = vars(factor(mt_id,levels=prototype_labels_extended)))+
  theme(axis.text=ggplot2::element_blank(),axis.ticks=ggplot2::element_blank())
```



Map trajectories onto prototypes

```
mt_data <- mt_spatialize(mt_data)
mt_data <- mt_map(mt_data,prototypes = mt_prototypes_ext,
  save_as="measures", grouping_variables = "group")
```

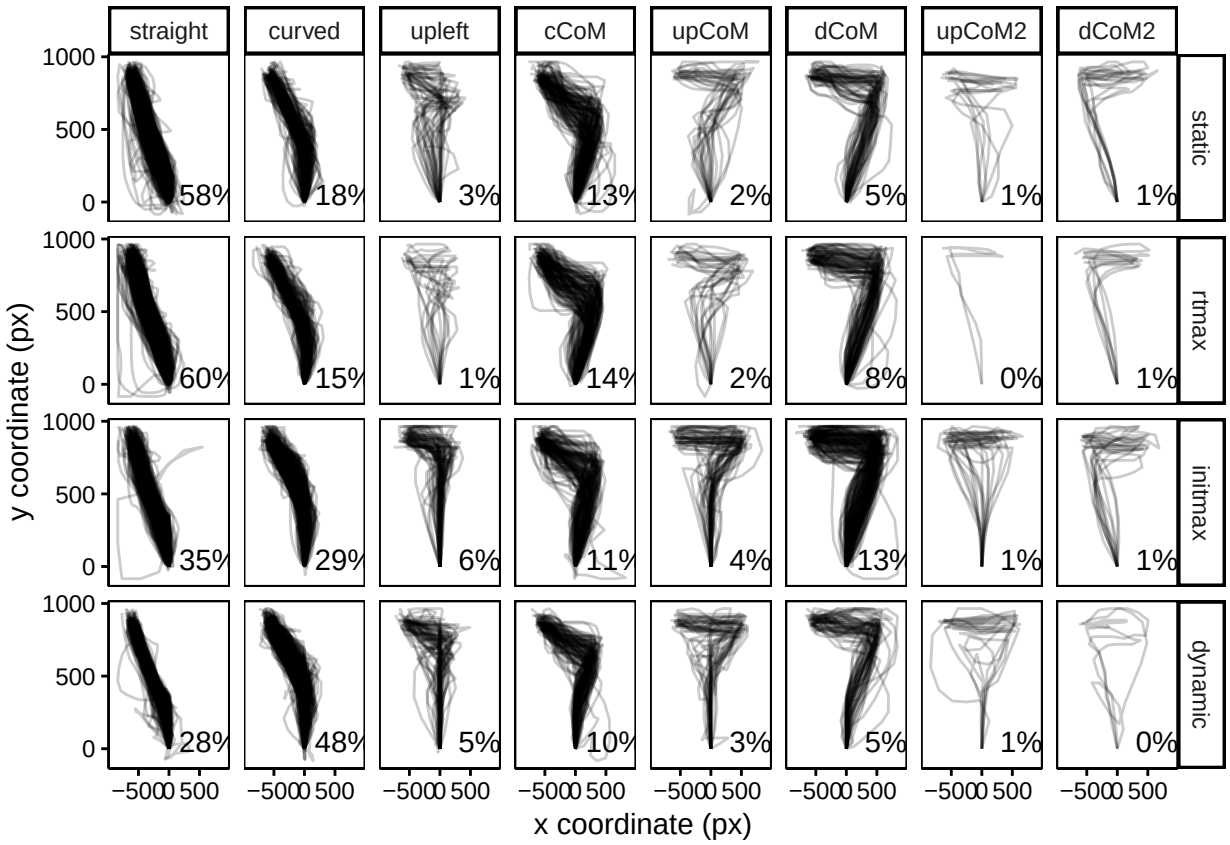
```
## Warning in create_results(data = data, results = results[, -1], use =
## use, : Columns of same name already exist and have been replaced
```

```
# Create variable that contains all prototypes in increasing order
mt_data$data$prototype_label <- factor(mt_data$measures$prototype_label,
  levels=prototype_labels_extended)

# Create variable that sets "up" prototypes equal to their curved equivalent
mt_data$data$prototype_label_red <- factor(mt_data$measures$prototype_label,
  levels=c("straight","curved","upleft","cCoM","upCoM","dCoM","upCoM2","dCoM2"),
  labels=c("straight","curved","curved","cCoM","cCoM", "dCoM","dCoM2", "dCoM2"))
```

Classified trajectories per group

Relative frequencies

```
prototype_percentages <- mt_data$data %>%
  group_by(group,prototype_label) %>%
  summarise(n=n()) %>%
  mutate(Percent=paste(round(100*n/sum(n)),"%",sep=""))

mt_plot(mt_data, use = "sp_trajectories",
  x = "xpos", y = "ypos", facet_col = "prototype_label", facet_row="group",alpha=.2)+
  xlab("x coordinate (px)") + ylab("y coordinate (px)")+
  geom_text(data=prototype_percentages,aes(label=Percent),x=650,y=50)+
  scale_y_continuous(breaks=c(0,500,1000))+
  coord_cartesian(xlim=c(-900,900))
```



Chi-squared test

```
chisq.test(with(mt_data$data,table(group, prototype_label)))
```

```
##
##  Pearson's Chi-squared test
##
## data:  with(mt_data$data, table(group, prototype_label))
## X-squared = 580.8, df = 21, p-value < 2.2e-16
```

Classified trajectories per group X typicality condition

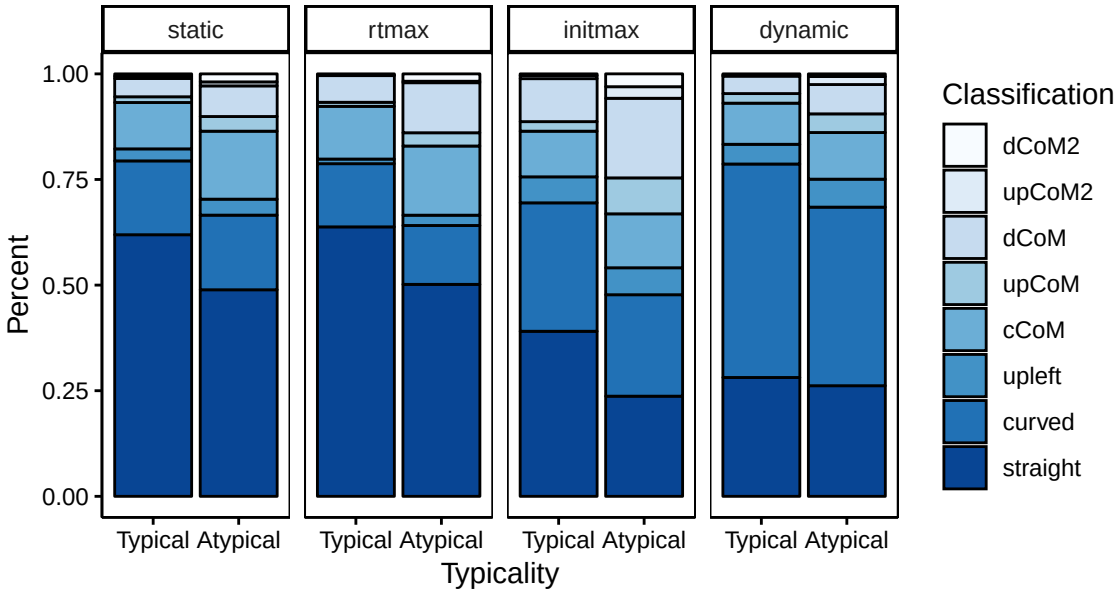
Relative frequencies

```
rel_freq_agg <- mt_data$data %>%
  group_by(group,Typicality,prototype_label) %>%
  summarise(n=n()) %>%
  mutate(Percent=n/sum(n), Percent_rounded = round(Percent,2))

spread(rel_freq_agg[,-c(4:5)],"prototype_label","Percent_rounded",fill = 0) %>%
  as.data.frame()
```

```
##     group Typicality straight curved upleft cCoM upCoM dCoM upCoM2 dCoM2
## 1  static    Typical     0.62   0.17   0.03 0.11  0.01 0.04   0.01  0.01
## 2  static   Atypical     0.49   0.18   0.04 0.16  0.03 0.07   0.01  0.02
## 3   rtmax    Typical     0.64   0.15   0.01 0.12  0.01 0.06   0.00  0.00
```

```
## 4   rtmax   Atypical     0.50   0.14   0.02 0.16  0.03 0.12   0.00  0.02
## 5 initmax    Typical     0.39   0.30   0.06 0.11  0.02 0.10   0.01  0.00
## 6 initmax   Atypical     0.24   0.24   0.06 0.13  0.09 0.19   0.03  0.03
## 7 dynamic    Typical     0.28   0.51   0.05 0.10  0.02 0.04   0.00  0.00
## 8 dynamic   Atypical     0.26   0.42   0.07 0.11  0.04 0.07   0.02  0.01
```

```
ggplot(rel_freq_agg,aes(x=Typicality,y=Percent,fill=forcats::fct_rev(prototype_label)))+
  geom_bar(stat="identity",color="black")+
  scale_fill_brewer(type="seq",name="Classification")+
  facet_grid(.~group)
```



Ordinal mixed regressions (treating all prototypes as ordered)

```
summary(clmm(prototype_label~Typicality*group+(1|subject_nr),data=mt_data$data))
```

```
## Cumulative Link Mixed Model fitted with the Laplace approximation
##
## formula: prototype_label ~ Typicality * group + (1 | subject_nr)
## data:    mt_data$data
##
##  link  threshold nobs logLik   AIC      niter      max.grad cond.H
##  logit flexible  4263 -5931.81 11893.62 2004(8020) 1.06e-02 1.3e+03
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  subject_nr (Intercept) 0.6844   0.8273
## Number of groups:  subject_nr 245
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## Typicality1                0.6978     0.1363   5.121 3.04e-07 ***
## grouprtmax                 0.0528     0.1825   0.289   0.7724
## groupinitmax               1.0524     0.1756   5.994 2.05e-09 ***
## groupdynamic               0.7566     0.1774   4.266 1.99e-05 ***
```

```
## Typicality1:grouprtmax     0.1595     0.1999   0.797   0.4252
## Typicality1:groupinitmax   0.2958     0.1825   1.621   0.1050
## Typicality1:groupdynamic  -0.4217     0.1796  -2.348   0.0189 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                 Estimate Std. Error z value
## straight|curved   0.1155     0.1290   0.895
## curved|upleft     1.5138     0.1314  11.524
## upleft|cCoM       1.7538     0.1320  13.288
## cCoM|upCoM        2.6757     0.1363  19.635
## upCoM|dCoM        2.9830     0.1386  21.520
## dCoM|upCoM2       4.9358     0.1794  27.513
## upCoM2|dCoM2      5.6186     0.2157  26.052
```

Ordinal mixed regressions (treating "up" and curved prototypes equal)

```
summary(clmm(prototype_label_red~Typicality*group+(1|subject_nr),data=mt_data$data))
```

```
## Cumulative Link Mixed Model fitted with the Laplace approximation
##
## formula: prototype_label_red ~ Typicality * group + (1 | subject_nr)
## data:    mt_data$data
##
##  link  threshold nobs logLik   AIC      niter      max.grad cond.H
##  logit flexible  4263 -5096.38 10216.75 1237(4952) 3.18e-03 2.0e+02
##
## Random effects:
##  Groups     Name        Variance Std.Dev.
##  subject_nr (Intercept) 0.6424   0.8015
## Number of groups:  subject_nr 245
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## Typicality1                0.68474    0.13708   4.995 5.88e-07 ***
## grouprtmax                 0.06291    0.17880   0.352   0.7249
## groupinitmax               1.02895    0.17199   5.983 2.20e-09 ***
## groupdynamic               0.77149    0.17365   4.443 8.88e-06 ***
## Typicality1:grouprtmax     0.13950    0.20087   0.694   0.4874
## Typicality1:groupinitmax   0.26786    0.18405   1.455   0.1456
## Typicality1:groupdynamic -0.44088    0.18115  -2.434   0.0149 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##                 Estimate Std. Error z value
## straight|curved   0.1281     0.1263   1.014
## curved|cCoM       1.7488     0.1294  13.517
## cCoM|dCoM         2.9604     0.1360  21.774
## dCoM|dCoM2        4.9059     0.1772  27.678
```