

# Tutorial analyses

*Pascal J. Kieslich, Felix Henninger, Dirk U. Wulff, Jonas M. B. Haslbeck, & Michael Schulte-Mecklenbeck*

## Preparations

### Load libraries

```
library(mousetrap)
library(ggplot2)
library(dplyr)
```

### 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)
  ))

options(width=90)
```

## Import

```
mt_data <- mt_import_mousetrap(KH2017_raw)
```

## Preprocessing

### Spatial transformations

```
# Remap trajectories
mt_data <- mt_remap_symmetric(mt_data)

# Align trajectories to common start position
mt_data <- mt_align_start(mt_data, start=c(0,0))
```

### Resampling

```
# Time-normalize trajectories
mt_data <- mt_time_normalize(mt_data)
```

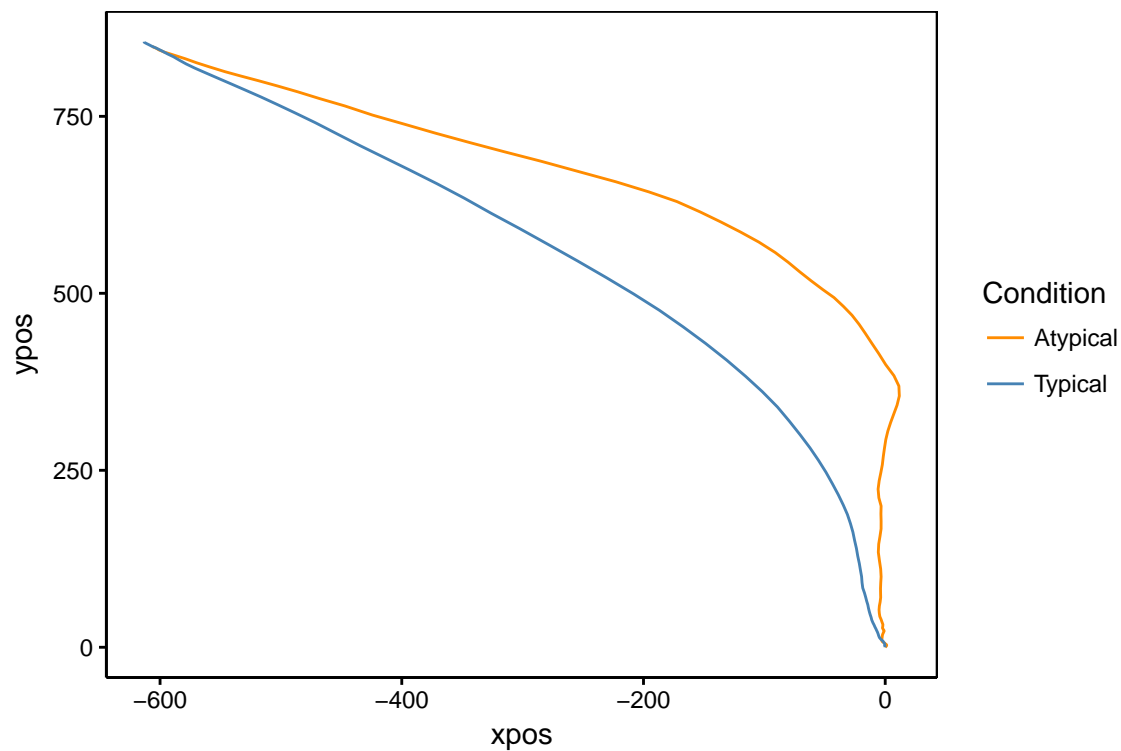
## Filtering

```
# Only keep trials with correct answers  
mt_data <- mt_subset(mt_data, correct==1)
```

## Analysis

### Aggregate trajectories

```
# Fig. 4  
mt_plot_aggregate(mt_data, use="tn_trajectories",  
  x="xpos", y="ypos",  
  color="Condition", subject_id="subject_nr")+  
  scale_color_manual(values=c("darkorange", "steelblue"))
```



### Calculate measures

```
# Calculate velocity and acceleration  
mt_data <- mt_derivatives(mt_data)  
  
# Calculate trial-level indices  
mt_data <- mt_measures(mt_data)  
  
# Calculate sample entropy  
mt_data <- mt_sample_entropy(mt_data, use="tn_trajectories")
```

## Curvature

### Aggregate analyses

```
# Aggregate MAD values per participant and condition
agg_mad <- mt_aggregate_per_subject(mt_data,
  use_variables="MAD", use2_variables="Condition",
  subject_id="subject_nr")

# Compare aggregated MAD values
t.test(MAD~Condition, data=agg_mad, paired=TRUE)

##
## Paired t-test
##
## data: MAD by Condition
## t = 6.731, df = 59, p-value = 7.706e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
## 120.5772 222.5949
## sample estimates:
## mean of the differences
## 171.5861

# Calculate descriptives
agg_mad %>%
  group_by(Condition) %>%
  summarise_at("MAD", .funs=c("mean", "sd")) %>%
  as.data.frame()

## Condition mean sd
## 1 Atypical 343.7954 218.6368
## 2 Typical 172.2093 110.8397
```

### Trial level analyses

```
# Create data.frame that contains the
# trial variables and mouse-tracking indices
results <- merge(mt_data$data, mt_data$measures, by="mt_id")

# Load afex package
library(afex)

# Run linear mixed model with Condition as a fixed effect
# and a random intercept and slope per participant
mixed(MAD ~ (1+Condition|subject_nr)+Condition, data=results)

## Fitting one lmer() model. [DONE]
## Calculating p-values. [DONE]

## Mixed Model Anova Table (Type 3 tests, KR-method)
##
## Model: MAD ~ (1 + Condition | subject_nr) + Condition
## Data: results
```

```
##      Effect      df      F p.value
## 1 Condition 1, 58.38 45.05 *** <.0001
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '+' 0.1 ' ' 1
```

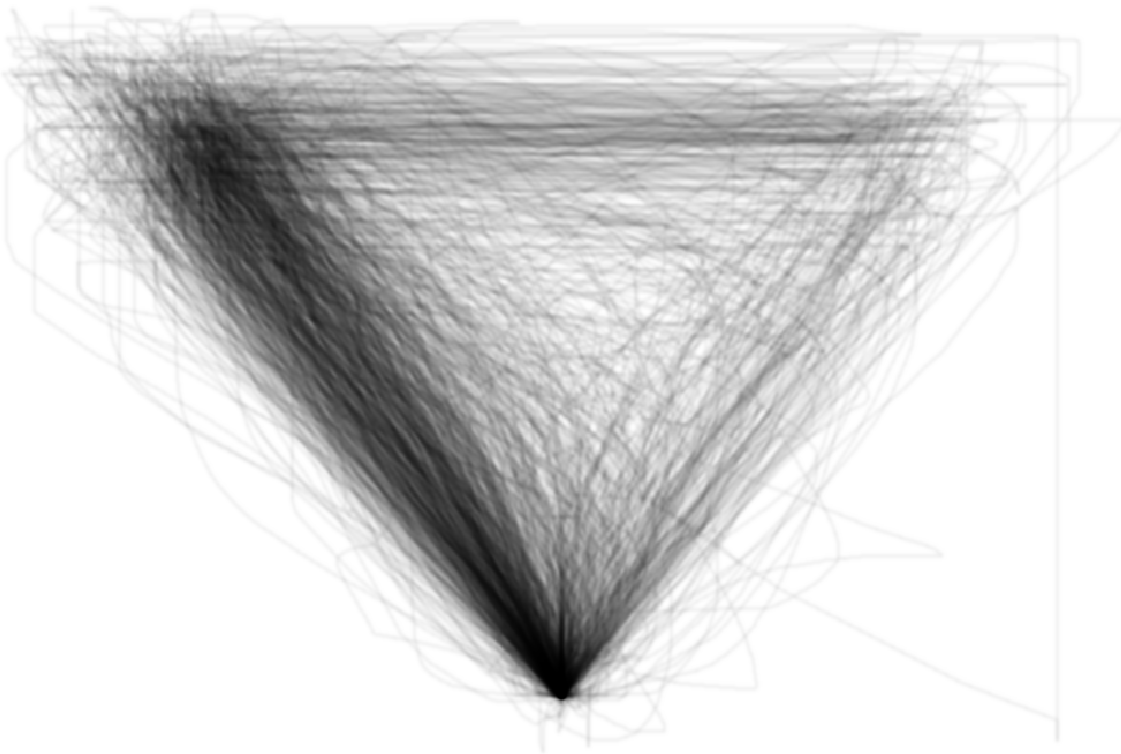
## Trajectory types

### Heatmap of individual trajectories

Fig. 5, top

```
mt_heatmap(mt_data,
  xres=1000,
  colors=c("white","black"))
```

```
## spatializing trajectories
## calculate image
## smooth image
## enhance image by 33.6
## creating heatmap: 1000 x 613 px
```



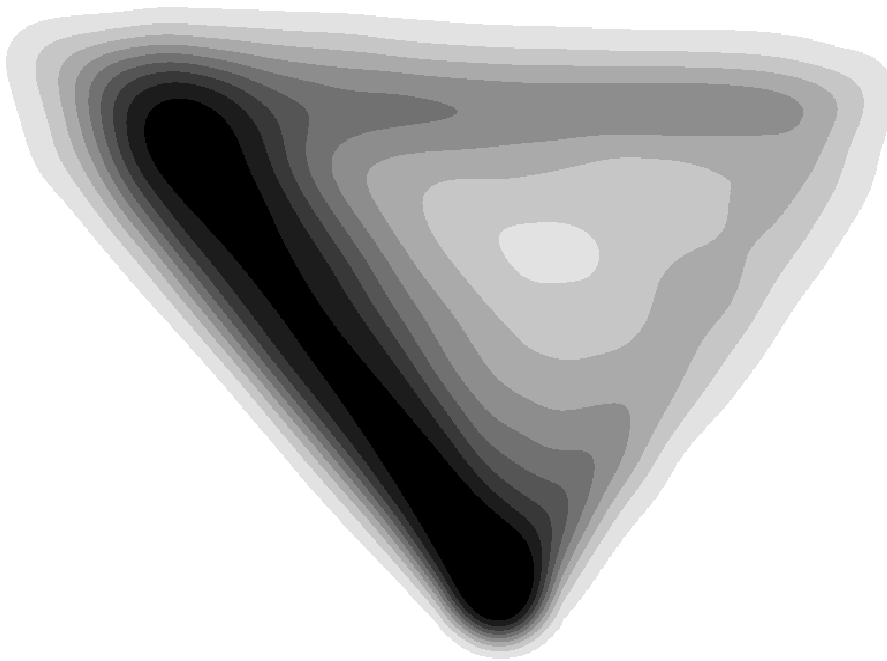
```
## heatmap created in 3s
```

## Smoothed heatmap

Fig. 5, middle

```
mt_heatmap(mt_data,  
           xres=1000,  
           colors=c("white", "black"),  
           smooth_radius=20,  
           n_shades=10,  
           mean_image=0.2)
```

```
## spatializing trajectories  
## calculate image  
## smooth image  
## enhance image by 4.1  
## creating heatmap: 1000 x 642 px
```



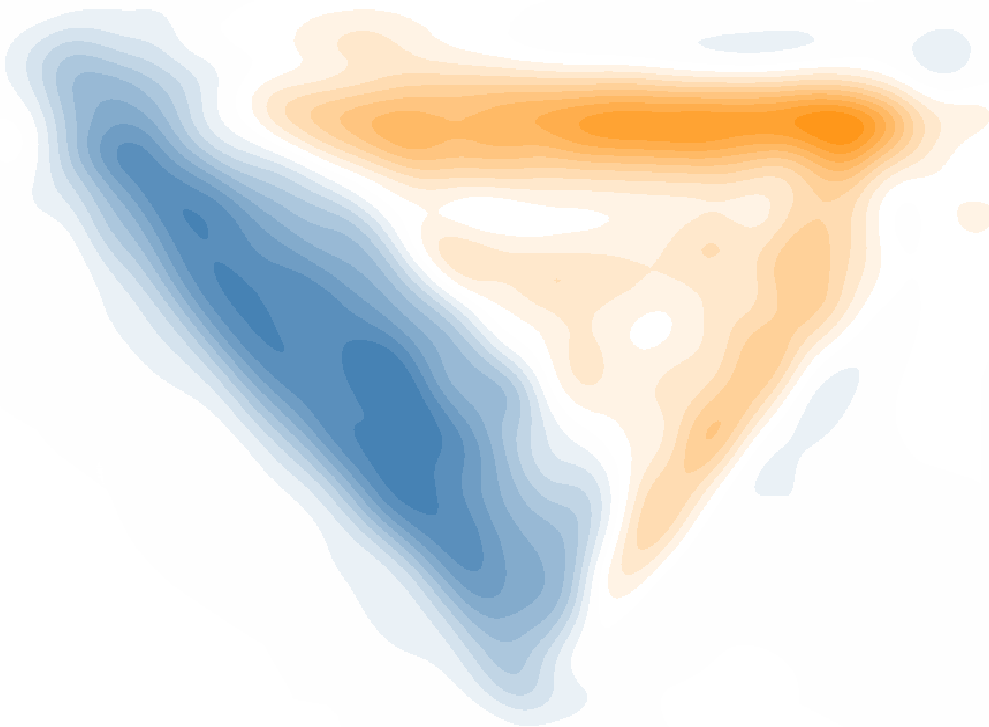
```
## heatmap created in 33s
```

## Difference of heatmaps between conditions

Fig. 5, bottom

```
mt_diffmap(mt_data,  
  xres=1000,  
  condition=mt_data$data$Condition=="Typical",  
  colors=c("steelblue","white","darkorange"),  
  smooth_radius=20,  
  n_shades=10)
```

```
## Determine joint bounds  
## Calculating heatmap for x  
## Calculating heatmap for y  
## smooth image  
## creating heatmap: 1000 x 611 px
```

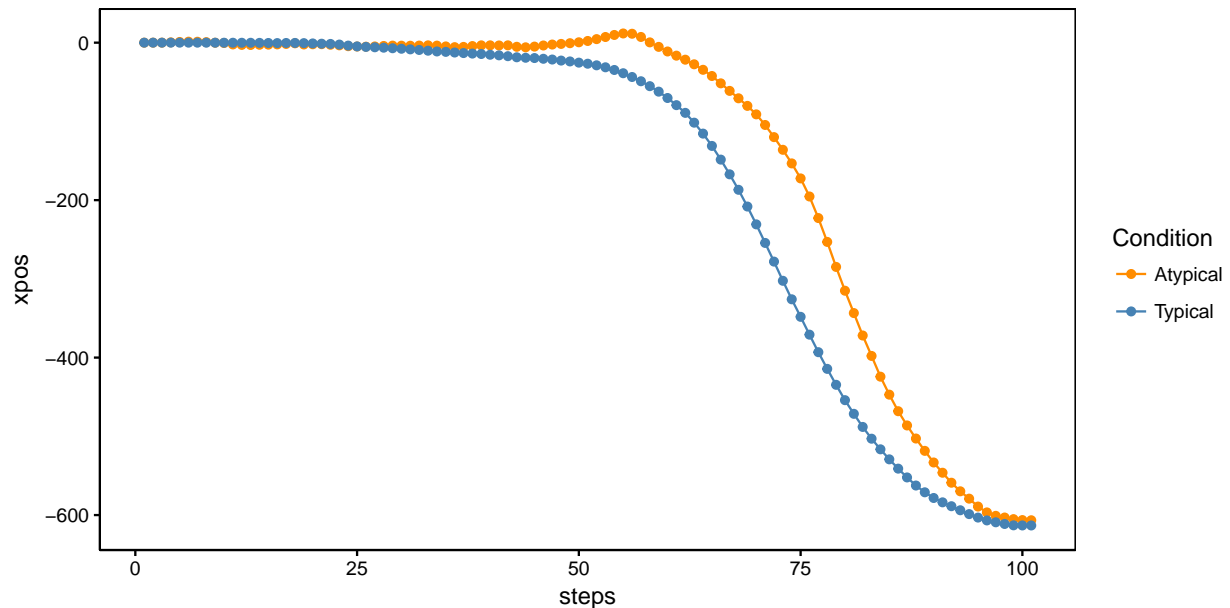


```
## heatmap created in 37s
```

## Temporal analyses

### Average x positions

```
# Plot aggregate time-normalized x-positions (Fig. 6)
mt_plot_aggregate(mt_data, use="tn_trajectories",
  x="steps", y="xpos", color="Condition",
  subject_id="subject_nr", points=TRUE)+
scale_color_manual(values=c("darkorange", "steelblue"))
```



```
# Aggregate time-normalized trajectories per condition
# separately per participant
av_tn_trajectories <- mt_aggregate_per_subject(mt_data,
  use="tn_trajectories", use2_variables="Condition",
  subject_id="subject_nr")
```

```
# Paired t-tests on coordinates
xpos_t_tests <-
  with(av_tn_trajectories,
    sapply(unique(steps), function(i){
      t.test(xpos[Condition=="Atypical" & steps==i],
        xpos[Condition=="Typical" & steps==i],
        paired = TRUE)$p.value})
  )
```

```
# Retrieve all significant t-tests
which(xpos_t_tests < .05)
```

```
## [1] 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81
## [29] 82 83 84 85 86 87 88 89 90 91 92 93 94 95
```

## Riverbed plot

Fig. 7

```
mt_plot_riverbed(mt_data, use="tn_trajectories",  
y="xpos", facet_col="Condition")
```

