

# Humpback Whales and Ship Noise

Fabian Blasch

05/23/2022



# 1 Data and Descriptive Statistics

```
# import data (this dataset is unfortunately not public)
openxlsx::read.xlsx("../Data/Humpback_Whales_Data.xlsx") -> dat_whale

# first a quick look at the missing values in the data
sapply(dat_whale, \(x) sum(is.na(x))) |> knitr::kable(col.names = "NAs")
```

	NAs
Individuum	0
Treatment	0
Szenario	0
ruhezeit	0
speed	5
Atem	0

```
# harmonize names
colnames(dat_whale) <- tolower(colnames(dat_whale))

# to numeric
lapply(dat_whale[, c("ruhezeit", "speed", "atem")], as.numeric) -> dat_whale[, c("ruhezeit", "speed", "atem")]

# to factor
lapply(dat_whale[, !(colnames(dat_whale) %in% c("ruhezeit", "speed", "atem"))],
       as.factor) -> dat_whale[, !(colnames(dat_whale) %in% c("ruhezeit", "speed", "atem"))]

# relevele
factor(dat_whale[, "szenario"],
       levels = c("Before", "During", "After")) -> dat_whale[, "szenario"]
factor(dat_whale[, "treatment"], c("Control", "Medium", "High")) -> dat_whale[, "treatment"]

# add log
within(dat_whale,{
  logspeed <- log(speed)
  atem[atem == 0] <- 0.001
  ruhezeit[ruhezeit == 0] <- 0.001
  logatem <- log(atem)
  logruhezeit <- log(ruhezeit)
  sqrtatem <- sqrt(atem)
  sqrtspeed <- sqrt(speed)
  sqrtruhezeit <- sqrt(ruhezeit * 100)
}) -> dat_whale

# first split into different intensities
dat_whale_intens <- split(dat_whale, dat_whale[, "treatment"])

# build formulas
formulae <- paste(c("ruhezeit", "speed", "atem"), "~", "szenario")

# max and min for plot y-axis
sapply(c(min, max), \(x){
```

```

sapply(dat_whale[, c("ruhezeit", "speed", "atem")], \(y) x(y, na.rm = TRUE))

}) -> ylims

# over szenarios
invis.Map(\(y, nom, lims){

  # safe for presentation
  # pdf(paste0("../Presentation/", nom, ".pdf"))

  # align
  par(mfrow = c(3, 1), mar = c(2, 4, 4, 2) + 0.1)

  # over treatment
  invis.Map(\(x, nom){

    # boxplots
    boxplot(as.formula(y), data = x,
            col = c("cornflowerblue", "deepskyblue4", "darkblue"),
            ylim = c(lims[1], lims[2]))

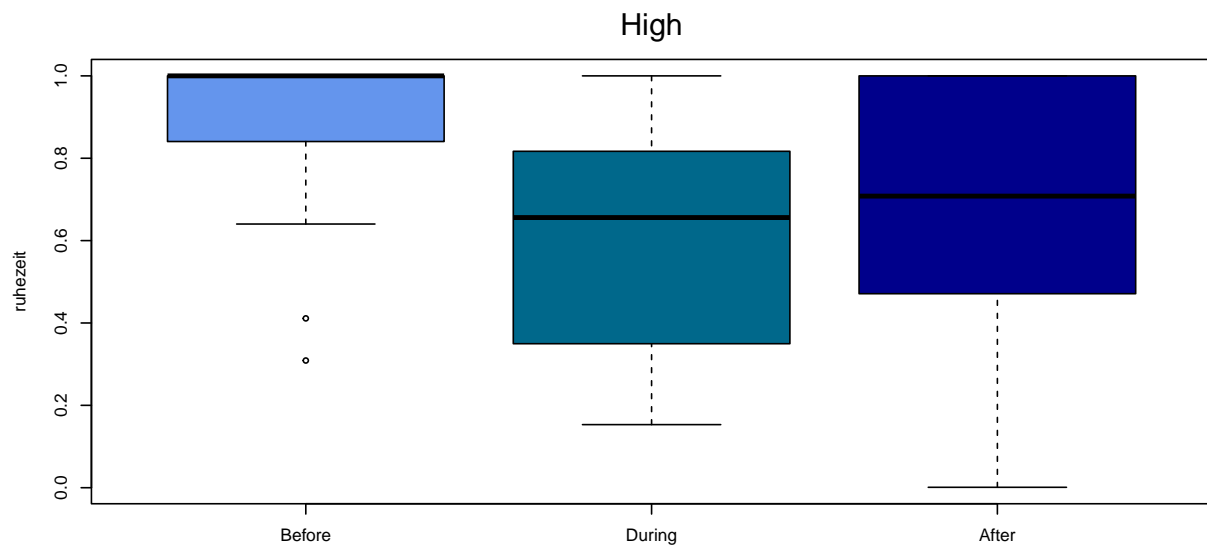
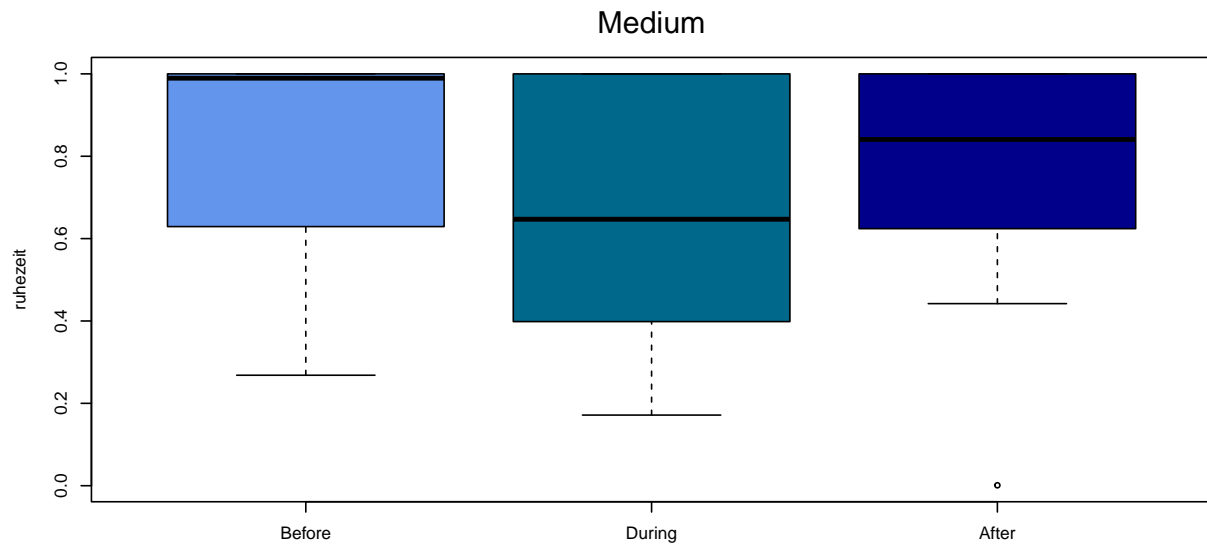
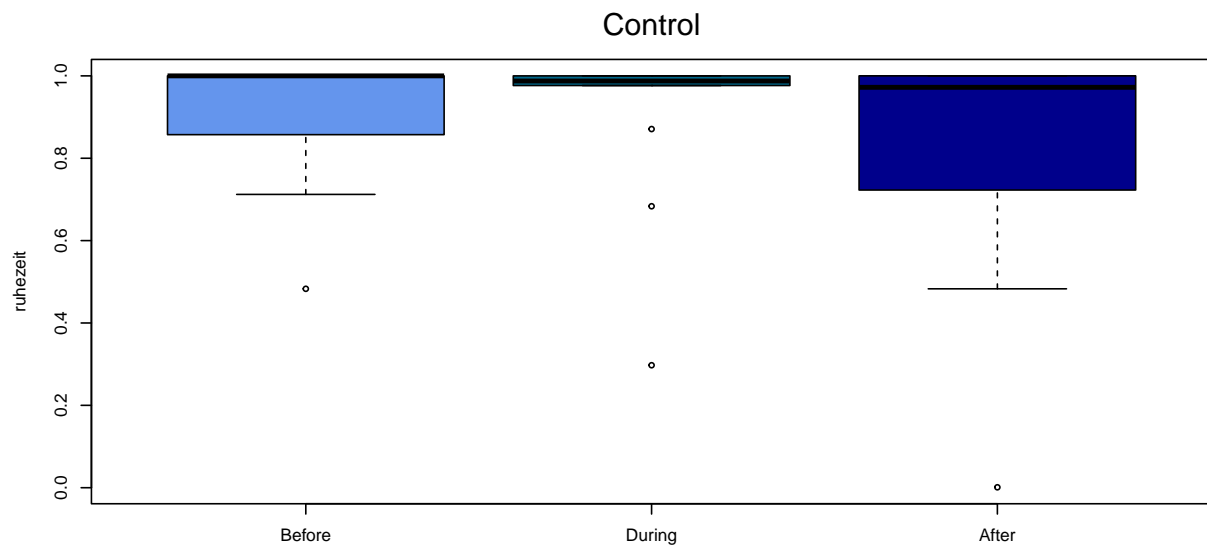
    # add label
    mtext(nom, side = 3, line = 1, cex = 1.2)

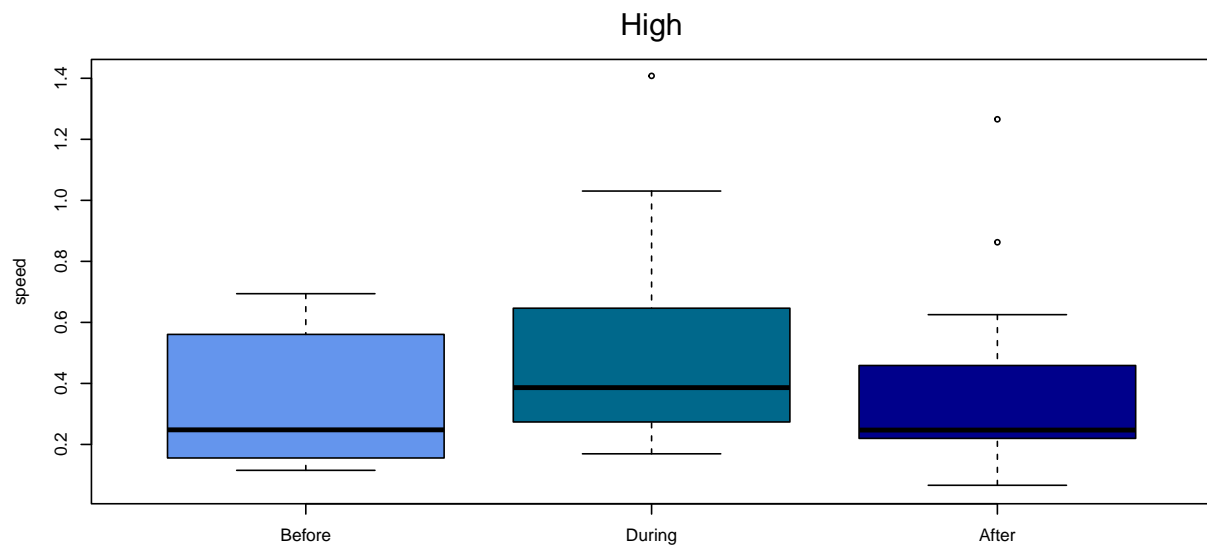
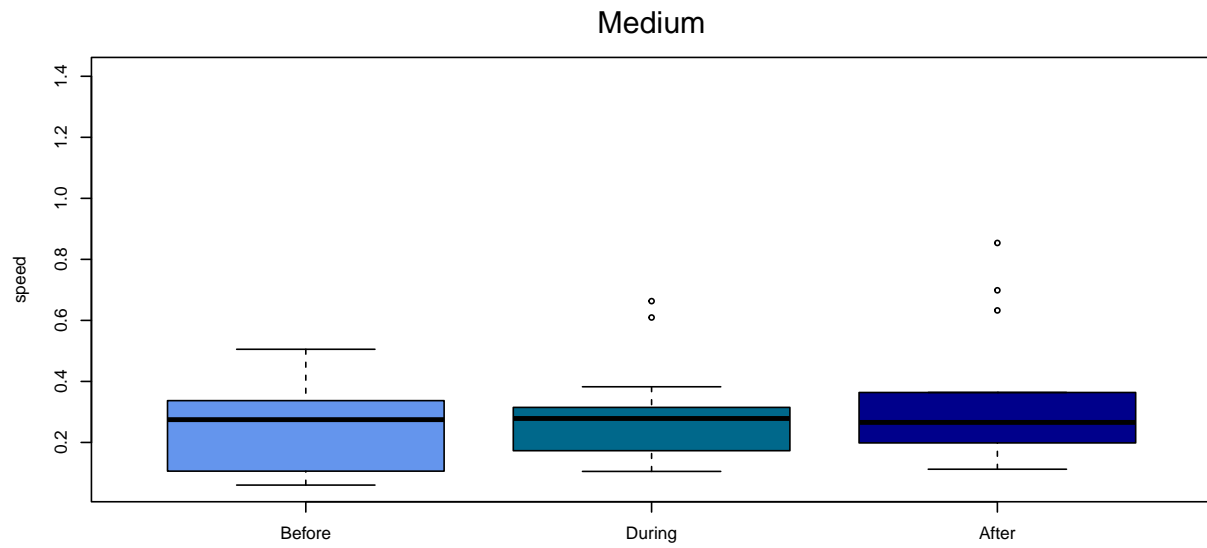
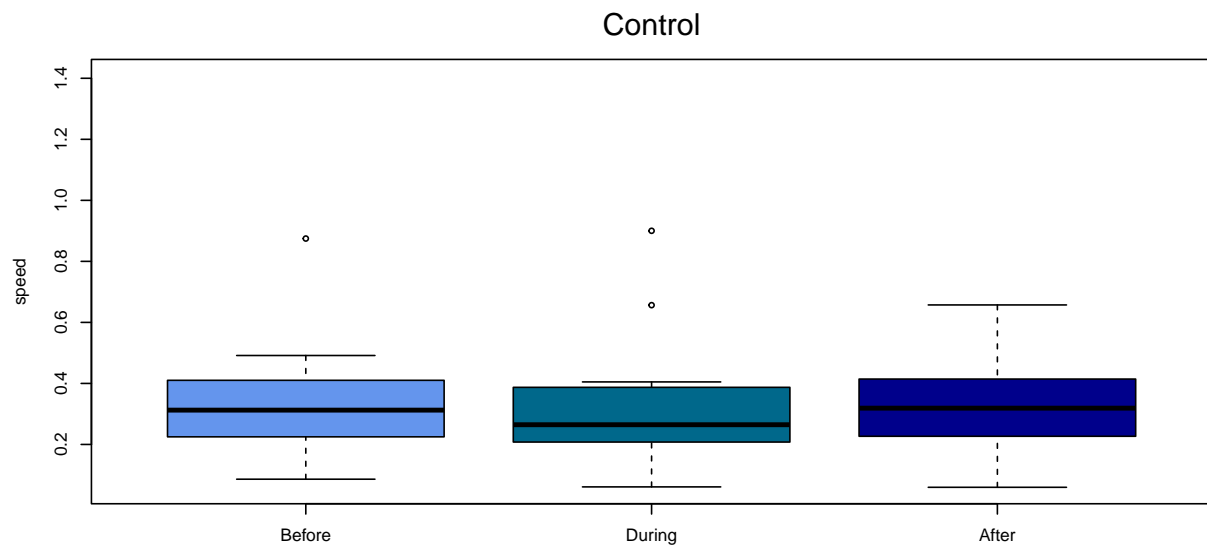
  }, dat_whale_intens, names(dat_whale_intens))

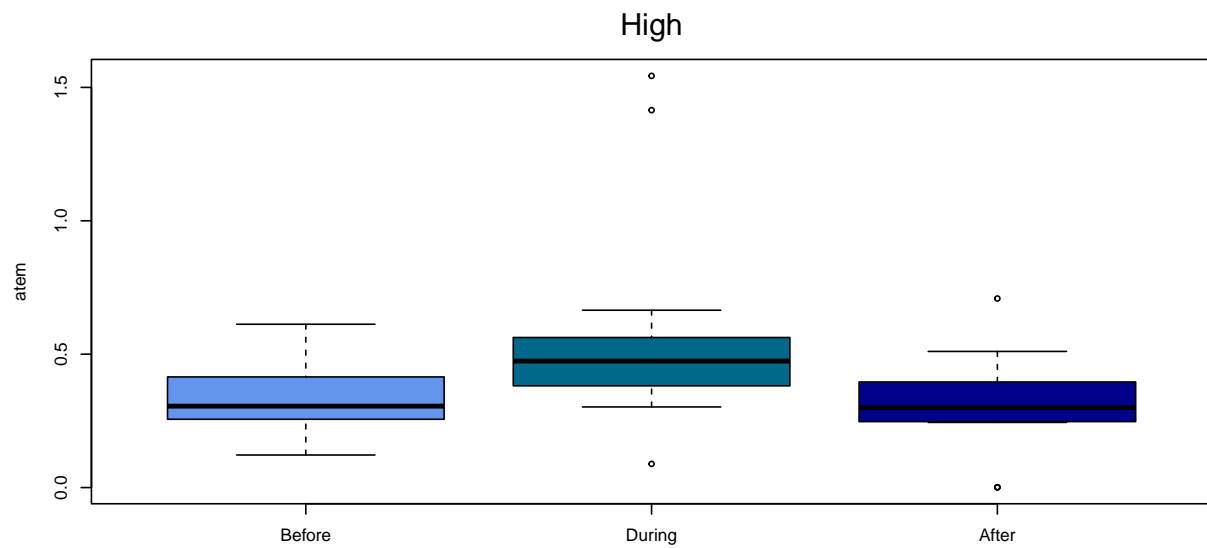
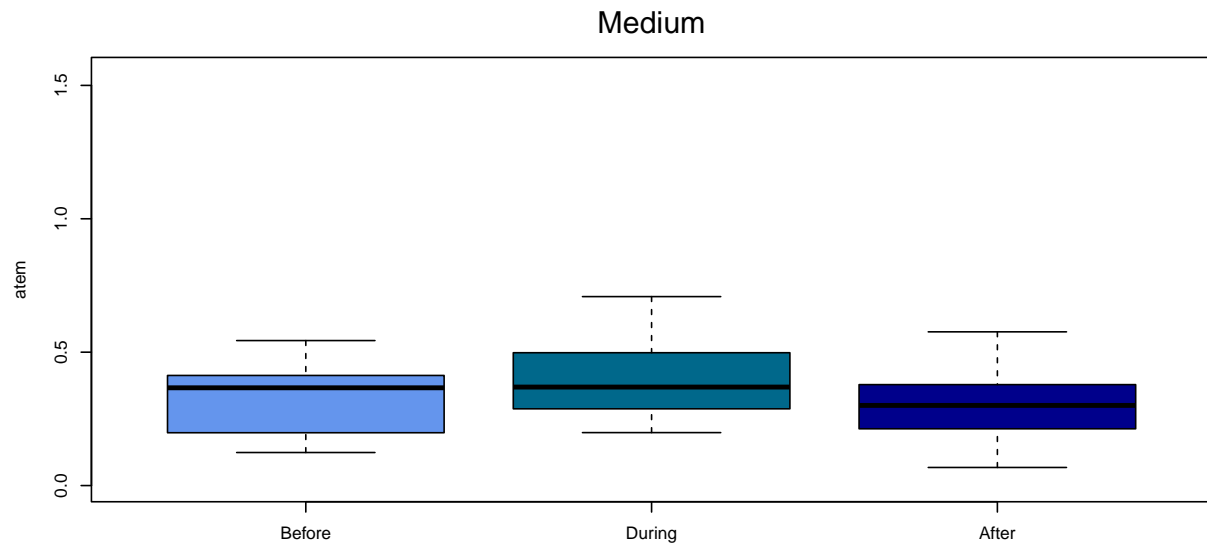
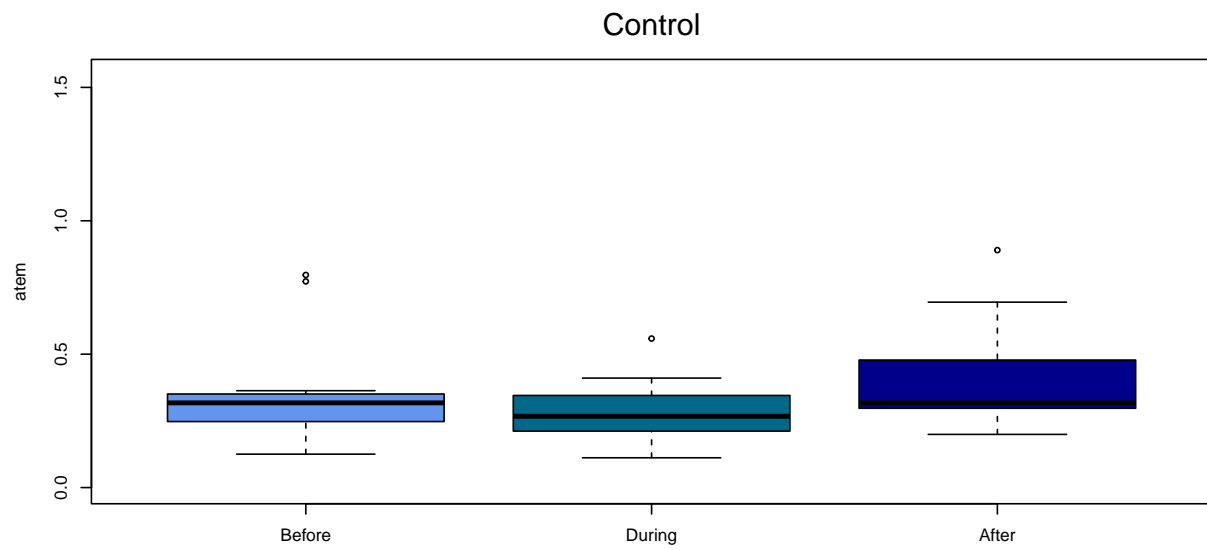
  # close graph. device
  # dev.off()

}, formulae, c("resting", "speed", "respatory"), ylims |> t() |> as.data.frame())

```





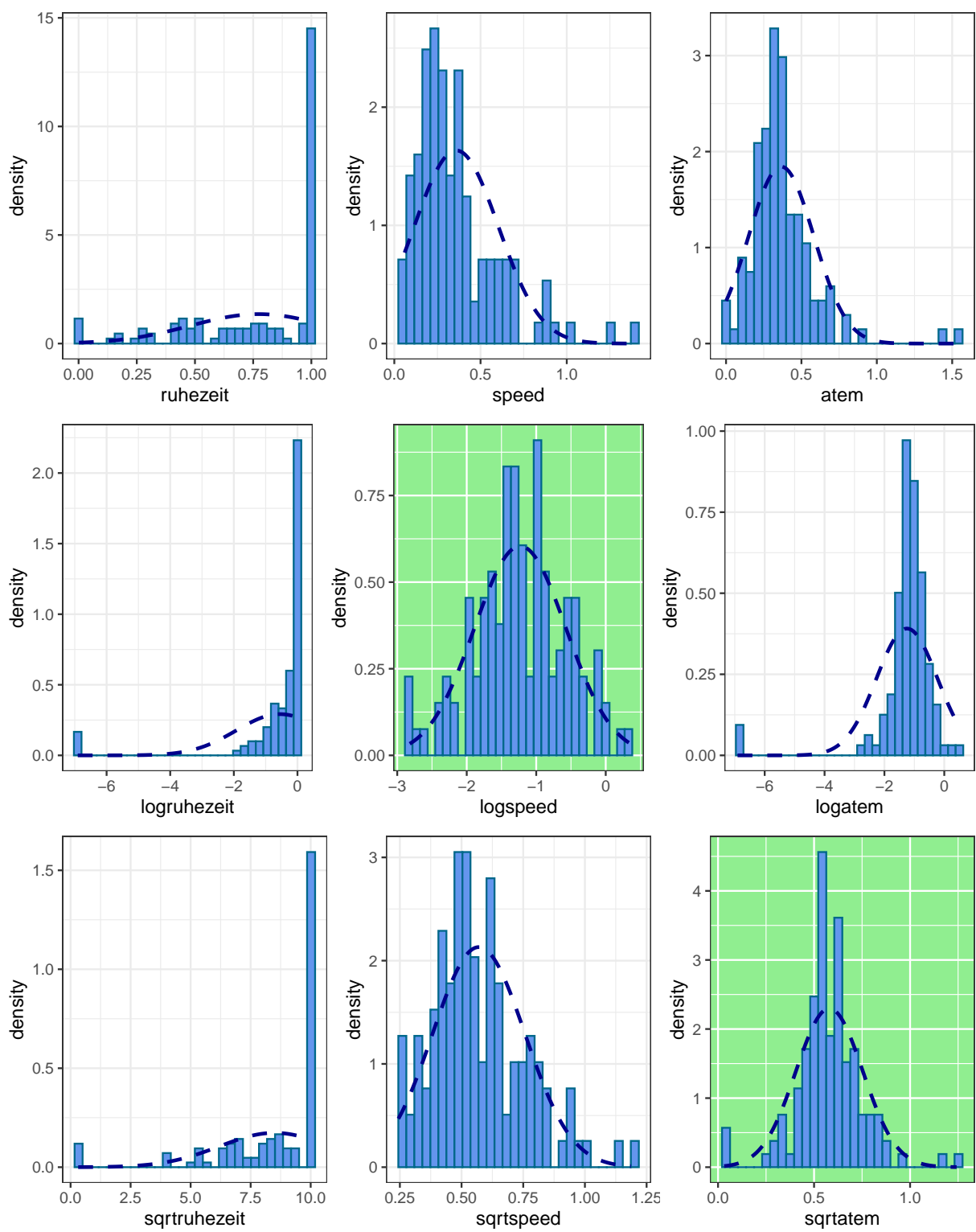


## 2 Distribution of Covariates

```
# vars
nom <- c("ruhezeit", "speed", "atem")
nom <- c(nom, paste0("log", nom), paste0("sqrt", nom))

# loop to generate plots
Map(\(x, bool) Dens_norm_plot(y = x, bg_alt = bool),
    nom, c(F, F, F, F, T, F, F ,F ,T)) -> plots

# display (remove useless bin width messages)
print((plots[[1]] + plots[[2]] + plots[[3]]) /
      (plots[[4]] + plots[[5]] + plots[[6]]) /
      (plots[[7]] + plots[[8]] + plots[[9]])) |>
  suppressWarnings() |>
  suppressMessages()
```





### 3 Model

```
# formulas
formulae_lmm <- c("speed ~ I(treatment) * I(szenario)",
                  "atem ~ I(treatment) * I(szenario)",
                  "logspeed ~ I(treatment) * I(szenario)",
                  "sqrtatem ~ I(treatment) * I(szenario)")

formulae_glmm <- "ruhezeit ~ I(treatment) * I(szenario)"

# comb
formulae_cmb <- list(formulae_lmm, formulae_glmm)

# fit models
Map(\(type, bool){

  lapply(type, \(x){

    if(bool){ # LMMs

      # fit LMM
      nlme::lme(as.formula(x), random = ~ 1 | individuum,
                data = dat_whale, na.action = na.omit,
                method = "REML") -> fit

      # summary
      list(fit,
            summary(fit))

    } else { # GLMMs

      # fit glmm PQL
      MASS::glmmPQL(as.formula(x), random = ~ 1 | individuum,
                    family = binomial(link = "logit"),
                    data = dat_whale) -> fit

      # summary
      list(fit,
            summary(fit))

    }

  }) |> setNames(type)

}, formulae_cmb, c(TRUE, FALSE)) |> setNames(c("LMM", "GLMM")) |>
  suppressMessages() |> suppressWarnings() -> models

# fits and summaries
LMM_fits <- lapply(models[[1]], "[[", 1)
LMM_summaries <- lapply(models[[1]], "[[", 2)
glmmPQL_fits <- lapply(models[[2]], "[[", 1)
glmmPQL_summaries <- lapply(models[[2]], "[[", 2)

# rebind fits for plotting
fits <- c(LMM_fits, glmmPQL_fits)
```

```
summaries <- c(LMM_summaries, glmmPQL_summaries)
```

## 4 Residual Diagnostics

```
# align
par(mfrow = c(5, 2), mar = c(2, 4, 4, 2) + 0.1)

# residual plots
invis.Map(\(x, nom){

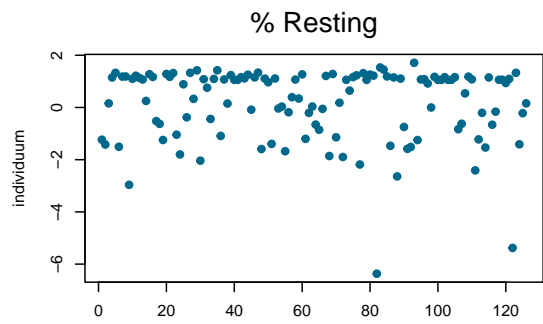
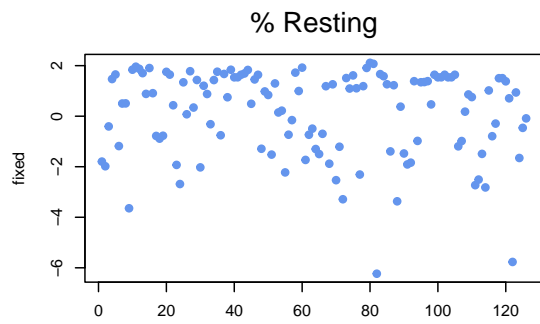
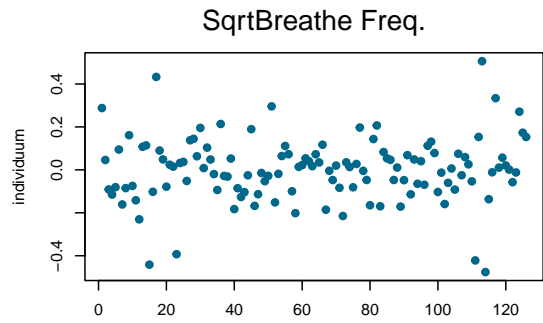
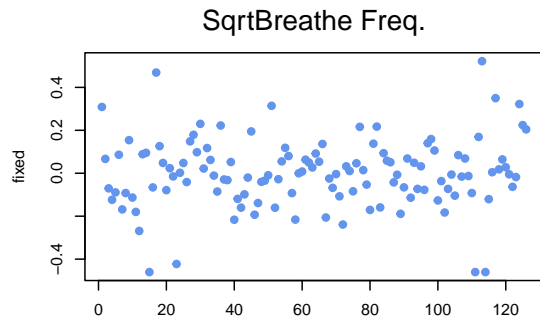
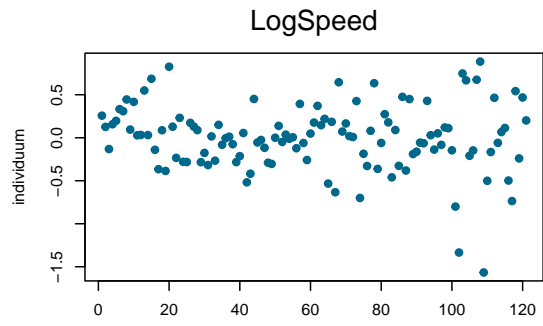
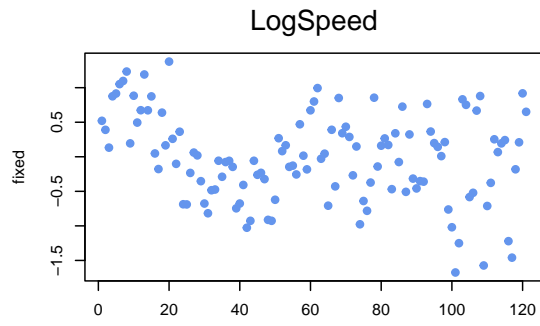
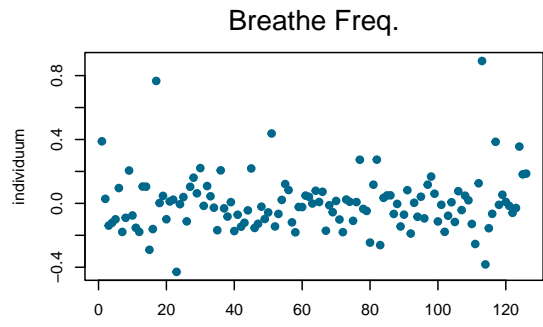
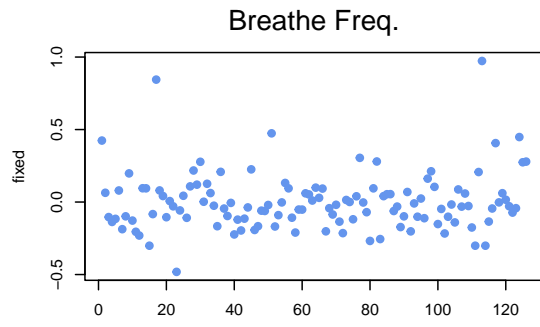
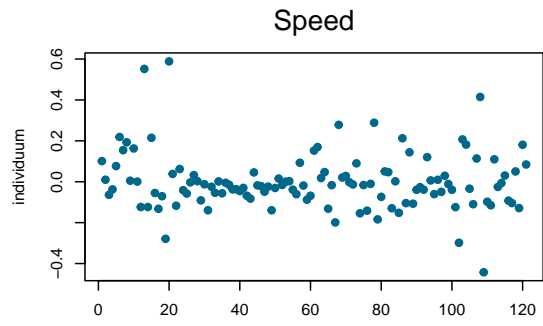
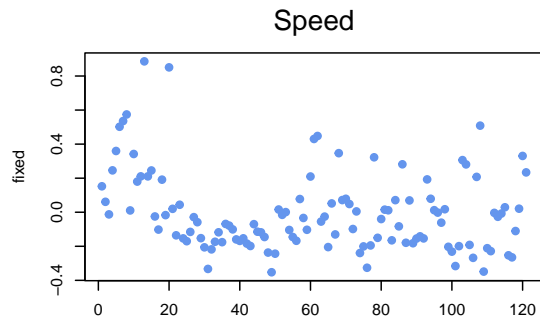
  invis.Map(\(y, col){

    # plots
    plot(x[["residuals"]][, y], type = "p", ylab = y, xlab = "", pch = 19,
         col = col)

    # label
    mtext(nom, side = 3, line = 1, cex = 1.2)

  }, c("fixed", "individuum"), c("cornflowerblue", "deepskyblue4"))

}, fits, c("Speed", "Breathe Freq.", "LogSpeed", "SqrtBreathe Freq.", "% Resting"))
```



```

# qqplots
par(mfrow = c(5, 2), mar = c(3.8, 4, 4, 2) + 0.1)

# residual plots
invisible.Map(\(x, nom){

  invisible.Map(\(y, col){

    # plots
    qqnorm(x[["residuals"]][, y], pch = 19, col = col, main = "")

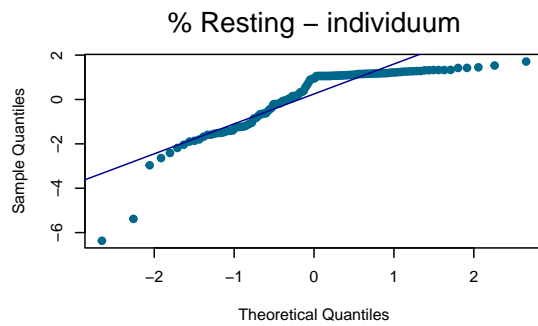
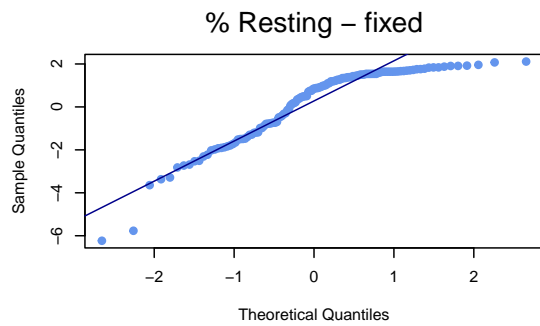
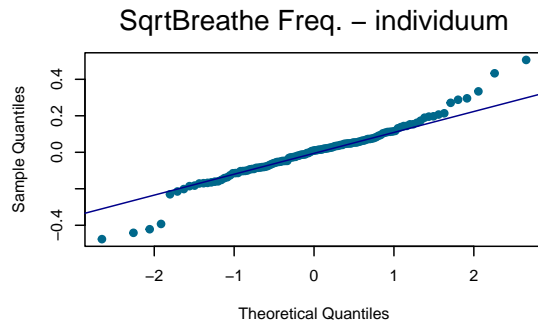
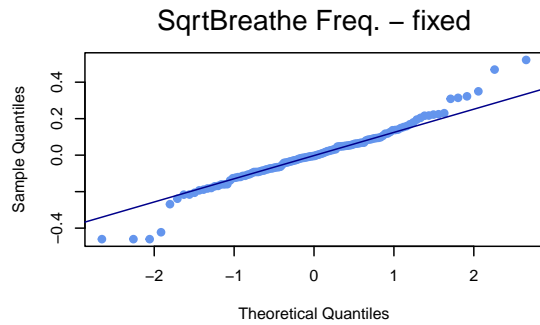
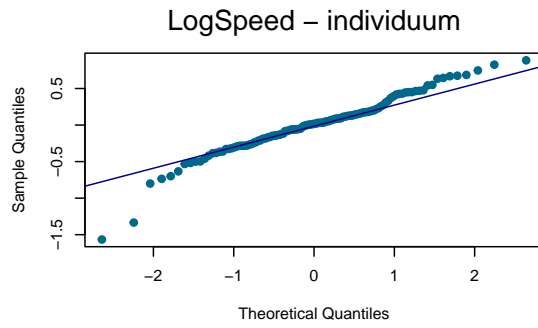
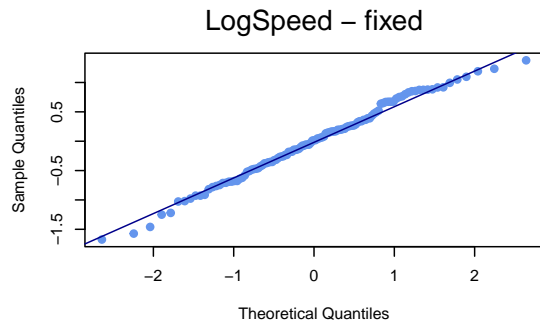
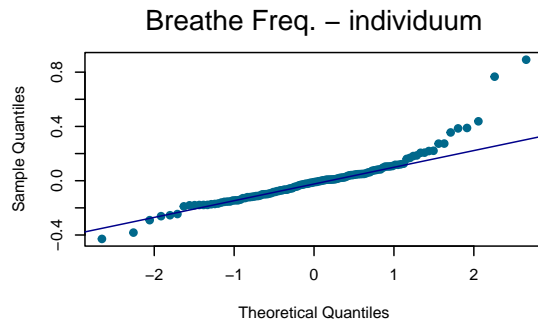
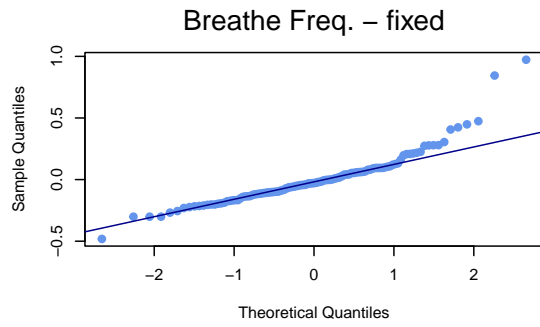
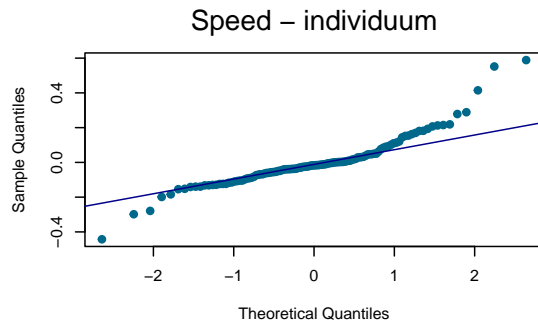
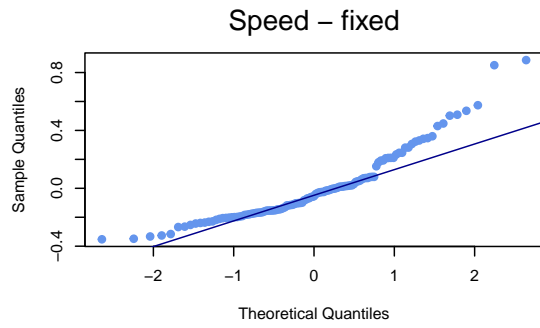
    # line
    qqline(x[["residuals"]][, y], col = "darkblue")

    # label
    mtext(paste(nom, "-", y), side = 3, line = 1, cex = 1.2)

  }, c("fixed", "individuum"), c("cornflowerblue", "deepskyblue4"))

}, fits, c("Speed", "Breathe Freq.", "LogSpeed", "SqrtBreathe Freq.", "% Resting"))

```



## 5 Coefficients and t-Tests

```
invis.lapply(names(summaries), \(x){
  # mods
  mod <- summaries[[x]][["tTable"]]

  # print table
  knitr::kable(mod)

}) |> setNames(names(summaries))
```

```
## $`speed ~ I(treatment) * I(szenario)`
##
##
## |
## | :-----: |-----: |---: |-----: |-----: |
## | (Intercept) | 0.3392015 | 0.0675357 | 73 | 5.0225532 | 0.0000035 |
## | I(treatment)Medium | -0.0796659 | 0.0950416 | 39 | -0.8382218 | 0.4070142 |
## | I(treatment)High | 0.0128286 | 0.0922713 | 39 | 0.1390310 | 0.8901411 |
## | I(szenario)During | -0.0132775 | 0.0659405 | 73 | -0.2013561 | 0.8409802 |
## | I(szenario)After | 0.0365231 | 0.0677611 | 73 | 0.5389981 | 0.5915282 |
## | I(treatment)Medium:I(szenario)During | 0.0571340 | 0.0932540 | 73 | 0.6126709 | 0.5419979 |
## | I(treatment)High:I(szenario)During | 0.1831087 | 0.0900919 | 73 | 2.0324658 | 0.0457448 |
## | I(treatment)Medium:I(szenario)After | 0.0556807 | 0.0941578 | 73 | 0.5913548 | 0.5561095 |
## | I(treatment)High:I(szenario)After | 0.0262800 | 0.0935384 | 73 | 0.2809546 | 0.7795400 |
##
## $`atem ~ I(treatment) * I(szenario)`
##
##
## |
## | :-----: |-----: |---: |-----: |-----: |
## | (Intercept) | 0.3487213 | 0.0568680 | 78 | 6.1321165 | 0.0000000 |
## | I(treatment)Medium | -0.0228728 | 0.0789743 | 39 | -0.2896229 | 0.7736389 |
## | I(treatment)High | -0.0161951 | 0.0776965 | 39 | -0.2084406 | 0.8359698 |
## | I(szenario)During | -0.0644547 | 0.0757239 | 78 | -0.8511814 | 0.3972742 |
## | I(szenario)After | 0.0673726 | 0.0757239 | 78 | 0.8897145 | 0.3763551 |
## | I(treatment)Medium:I(szenario)During | 0.1421530 | 0.1051600 | 78 | 1.3517785 | 0.1803532 |
## | I(treatment)High:I(szenario)During | 0.3022035 | 0.1034585 | 78 | 2.9210116 | 0.0045609 |
## | I(treatment)Medium:I(szenario)After | -0.0945670 | 0.1051600 | 78 | -0.8992675 | 0.3712780 |
## | I(treatment)High:I(szenario)After | -0.0978333 | 0.1034585 | 78 | -0.9456288 | 0.3472582 |
##
## $`logspeed ~ I(treatment) * I(szenario)`
##
##
## |
## | :-----: |-----: |---: |-----: |-----: |
## | (Intercept) | -1.2308608 | 0.1828628 | 73 | -6.7310605 | 0.0000000 |
## | I(treatment)Medium | -0.3267841 | 0.2573970 | 39 | -1.2695723 | 0.2117607 |
## | I(treatment)High | -0.0179156 | 0.2498382 | 39 | -0.0717087 | 0.9432002 |
## | I(szenario)During | -0.1078457 | 0.1807837 | 73 | -0.5965456 | 0.5526563 |
## | I(szenario)After | 0.0853729 | 0.1857559 | 73 | 0.4595971 | 0.6471713 |
## | I(treatment)Medium:I(szenario)During | 0.3393236 | 0.2556667 | 73 | 1.3272106 | 0.1885751 |
## | I(treatment)High:I(szenario)During | 0.5076296 | 0.2469975 | 73 | 2.0552010 | 0.0434380 |
```

```
## |I(treatment)Medium:I(szenario)After | 0.2628398| 0.2581038| 73| 1.0183495| 0.3118751|
## |I(treatment)High:I(szenario)After | 0.0214941| 0.2564104| 73| 0.0838269| 0.9334235|
##
## $`sqrtatem ~ I(treatment) * I(szenario)`
##
##
## | | Value| Std.Error| DF| t-value| p-value|
## |:-----:|-----:|---:|-----:|-----:|
## |(Intercept) | 0.5703680| 0.0458895| 78| 12.4291561| 0.0000000|
## |I(treatment)Medium | -0.0123334| 0.0637282| 39| -0.1935318| 0.8475469|
## |I(treatment)High | -0.0050724| 0.0626970| 39| -0.0809038| 0.9359322|
## |I(szenario)During | -0.0472330| 0.0622771| 78| -0.7584316| 0.4504783|
## |I(szenario)After | 0.0589025| 0.0622771| 78| 0.9458125| 0.3471650|
## |I(treatment)Medium:I(szenario)During | 0.1147202| 0.0864861| 78| 1.3264577| 0.1885589|
## |I(treatment)High:I(szenario)During | 0.2023393| 0.0850868| 78| 2.3780343| 0.0198531|
## |I(treatment)Medium:I(szenario)After | -0.0878070| 0.0864861| 78| -1.0152724| 0.3131150|
## |I(treatment)High:I(szenario)After | -0.1322060| 0.0850868| 78| -1.5537787| 0.1242856|
##
## $`ruhezeit ~ I(treatment) * I(szenario)`
##
##
## | | Value| Std.Error| DF| t-value| p-value|
## |:-----:|-----:|---:|-----:|-----:|
## |(Intercept) | 2.3808540| 0.6231463| 78| 3.8206985| 0.0002658|
## |I(treatment)Medium | -0.7865013| 0.7862894| 39| -1.0002696| 0.3233462|
## |I(treatment)High | -0.3502397| 0.8093376| 39| -0.4327486| 0.6675812|
## |I(szenario)During | -0.0267061| 0.8095529| 78| -0.0329887| 0.9737679|
## |I(szenario)After | -1.0226216| 0.7058604| 78| -1.4487589| 0.1514133|
## |I(treatment)Medium:I(szenario)During | -0.7751521| 0.9789670| 78| -0.7918062| 0.4308751|
## |I(treatment)High:I(szenario)During | -1.5056474| 0.9912816| 78| -1.5188896| 0.1328334|
## |I(treatment)Medium:I(szenario)After | 0.7143004| 0.9101265| 78| 0.7848364| 0.4349268|
## |I(treatment)High:I(szenario)After | -0.3409966| 0.9105983| 78| -0.3744753| 0.7090669|
```