

Datenauswertung

Jonas Schropp

25 November 2018

Benötigte Pakete:

```
library (dplyr)      # Datenbereinigung
library (readr)      # Einlesen und Schreiben der Daten
library (tidyr)      # Datenbereinigung
library (ggplot2)    # Visualisierungen
library (stringr)    # Generelle Textverarbeitung
library (tidytext)    # Generelle Textverarbeitung
library (text2vec)    # GloVe-Modellierung
library (lexicon)     # Stoppwörter
library (lme4)        # Mixed-Effects Modeling
library (caret)       # Datenstransformation
library (sjstats)     # Statistiken für Mixed-Effects Model
library (multcomp)    # Paarweise Vergleiche für Mixed-Effects ANOVA
library (nlopt)       # Optimierung für Mixed-Effects Modeling
library (lmerTest)    # p-Werte für Mixed-Effects Models
library (BBmisc)      # POMS-Standardisierung
library (quanteda)    # Generelle Textverarbeitung
library (lubridate)   # Daten und Zeiträume
library (knitr)       # Formatierung des Dokuments
```

Datenbereinigung (I)

Datensatz einlesen:

```
reddit_df <- read_rds(file.choose())
```

Im Datensatz erfasste Variablen:

- subreddit: Der Name des Subreddits, in dem gepostet wurde.
- author: Der Username des jeweiligen Verfassers eines Posts
- created_utc: Der Zeitpunkt, zu dem ein Post verfasst wurde (in Sekunden seit 01.01.1970)
- body: Der Kommentar in Textform

Einige der Kommentare und Autoren wurden nachträglich gelöscht. Leider ist es nicht möglich, festzustellen, ob diese *at random* entfernt wurden, oder beispielsweise aufgrund von Regelverletzungen.

Am einfachsten lassen sich die Kommentare mit *dplyr* entfernen:

```
reddit_df <- reddit_df %>%
  filter (body != "[removed]",
          author != "[deleted]",
          body != "[deleted]")
```

Einige der Kommentare innerhalb der *Subreddits* stammen nicht von echten Autoren, sondern von Programmen, meist mit bestimmten Aufgaben (z.B. das Korrigieren von Rechtschreibfehlern, oder die automatische Moderation von Kommentaren). Während es nicht möglich ist, mit absoluter Sicherheit festzustellen, welche Kommentare von echten Menschen verfasst wurden und welche nicht, kann zumindest der jeweilige *Automoderator* entfernt werden, sowie Autoren deren Namen *bot* oder *Bot* beinhaltet.

In diesem Schritt können auch die Autoren entfernt werden, welche nicht mindestens vier verschiedene Posts verfasst haben (vgl. Nonneke & Preece, 2000), um nur regelmäßige Poster im Datensatz zu behalten.

Dafür werden zuerst alle Autoren in einen eigenen Datensatz überführt, der sowohl deren Usernamen, als auch die Anzahl an Posts des Autoren enthält.

```
author_count <- reddit_df %>%
  group_by(subreddit) %>%
  count(author)

reddit_df <- left_join(reddit_df, author_count, on = c("subreddit", "author"))

bots <- str_subset(reddit_df$author, 'bot')
Bots <- str_subset(reddit_df$author, 'Bot')
reddit_df_bots <- reddit_df$author %in% bots | reddit_df$author %in% Bots

reddit_df_not_bots <- reddit_df$author[!reddit_df_bots]

reddit_df <- reddit_df %>%
  filter(author %in% reddit_df_not_bots,
         n >= 4,
         author != "AutoModerator")
```

GloVe-Modell

Anhand dieses vorläufig bereinigten Datensatzes von 31,674,175 Kommentaren kann das GloVe-Modell (Pennington et al., 2014) berechnet werden, das daraufhin zur Kodierung der Kommentare eingesetzt wird. Dieses wird frühzeitig berechnet, um eine möglichst große Zahl an Kommentaren einbeziehen zu können. Bis jetzt gibt es keine klaren Erkenntnisse, wie viele Texte/Worte mindestens notwendig sind, um sinntragende Wort-Vektor Repräsentationen zu erhalten, daher empfiehlt es sich, das Modell an einem möglichst großen Korpus zu trainieren.

Dafür wird in einem ersten Schritt jedem Kommentar eine ID (Variablenname: *comment_id*) zugeteilt:

```
reddit_df$comment_id <- 1:ncol(reddit_df)
```

Daraufhin wird eine Liste mit Stoppwörtern (Wörter, die bei der Textverarbeitung nicht beachtet werden, da sie sehr häufig auftreten und keine Relevanz für die Erfassung des Dokumentinhalts besitzen, z.B. “and”, “the”, oder “of”) erstellt, um die Worteinbettungen zu verbessern. Diese Liste basiert auf der Stoppwortliste *sw_loughran_mcdonald_long*, die im Paket *lexicon* zur Verfügung steht (Loughran & McDonald, 2016) und in der automatisierten Textverarbeitung weite Verbreitung gefunden hat. Dieses Verfahren führte bereits bei anderen Studien zu besseren Ergebnissen des GloVe-Modells (Lison & Kutuzov, 2017). Nicht miteinbezogen wurden Personalpronomen, da diese möglicherweise eine Orientierung von Kommentaren hin zur Ingroup oder Outgroup repräsentieren können.

```
contractions <- c("i'm", "we're", "you're", "they're", "he's", "she's", "it's",
                  "i've", "we've", "you've", "they've",
                  "i'd", "we'd", "you'd", "they'd", "he'd", "she'd", "it'd",
                  "i'll", "we'll", "you'll", "they'll", "he'll", "she'll", "it'll")

pronouns <- tolower(c(pos_df_pronouns$pronoun, contractions))

stopwords <- tolower(setdiff(sw_loughran_mcdonald_long, pronouns))
```

Die Berechnung der Wort-Vektoren erfolgt mit dem R-Paket *text2vec 0.5.1* (Selivanov, 2018).

Im ersten Schritt wird eine Iterator-Funktion erstellt, die den Textkorpus Stückweise bearbeitet.

```
it <- itoken(iterable = reddit_df$body,
            ids = reddit_df$comment_id,
            preprocessor = tolower,
            tokenizer = word_tokenizer,
            n_chunks = 500)
```

Iterativ wird das Vokabular der untersuchten Texte zusammengefasst und von den Stoppwörtern bereinigt. Daraufhin werden Worte aus dem Vokabular entfernt, die nicht mindestens 25 mal im gesamten Datensatz vorkommen. Dieser Schritt dient dem Entfernen von seltenen Ausdrücken, deren Bedeutung aufgrund dessen nicht adäquat abgebildet werden kann. Darüber hinaus können so falsch geschriebene Worte, Weblinks und Ähnliches ausgeschlossen werden. Wie häufig ein Wort vorkommen muss, um bedeutungstragende Worteinbettungen zu berechnen, hängt vom individuellen Datensatz ab, wobei die Zahl meist zwischen 10 und 100 liegt (Lison & Kutuzov, 2017).

```
vocab <- create_vocabulary(it, stopwords = stopwords)
```

```
pruned_vocab <- prune_vocabulary(vocab, term_count_min = 25L)
```

Das so entstandene Vokabular von 2,897,705 Worten verringert sich durch diesen Schritt auf 122,072 einzelne Worte.

Aus dem auf diese Weise bereinigten und vectorisierten Vokabular wird eine dünnbesetzte *Term-Dokument-Matrix* (TCM) erstellt. Bei der Erstellung werden symmetrisch 10 Worte vor und nach dem betreffenden Ausdruck miteinbezogen. Nach Goldberg (2016) führen größere Fenster zu Worteinbettungen, die bessere Ergebnisse in Wortanalogie Aufgaben erzielen, wobei verhältnismäßig kleinere Fenster Wortähnlichkeiten besser abbilden können. Das hier verwendete Fenster von 10 Unigrammen bezieht einen möglichst großen Teil des jeweiligen Kommentars mit ein und entspricht der Fenstergröße, die bei Lison und Kutuzov (2017) in einem vergleichbaren Datensatz die besten Ergebnisse im Wortanalogietest erzielte.

```
reddit_tcm <- create_tcm(it,
                        vectorizer,
                        skip_grams_window = 10L,
                        skip_grams_window_context = c("symmetric"))
```

Anhand dieser TCM kann daraufhin das GloVe-Modell berechnet werden. Entsprechend der Originalstudie von Pennington et al. (2014) basiert das Modell auf 50 Iterationen mit einem x_max von 100, sowie einer Vektorenlänge von 100.

```
glove_model <- GlobalVectors$new(word_vectors_size = 100L,
                                vocabulary = pruned_vocab,
                                x_max = 100L)
```

```
vectors_main <- glove_model$fit_transform(reddit_tcm, n_iter = 50)
```

Die für die Kodierung verwendeten Wort-Vektoren basieren entsprechend der Empfehlungen von Pennington et al. (2014) auf den aggregierten Haupt- und Komponentenvektoren.

```
vectors_components <- glove_model$components
```

```
reddit_glove_vectors = (vectors_main + t(vectors_components))
```

Die Qualität der so ermittelten Wortvektoren kann anhand der Analogiefragen von Mikolov et al. (2013) abgeschätzt werden.

```
model_word_set <- pruned_vocab$term
```

```
setwd("~/MASTERARBEIT/Daten_Masterarbeit/Dictionary Creation")
```

```
questions_file <- ('questions-words.txt')
qlst <- prepare_analogy_questions(questions_file, model_word_set)
```

```
## INFO [2018-12-05 11:49:53] 14046 full questions found out of 19544 total
```

```
res <- check_analogy_accuracy(questions_list = qlst,
                             m_word_vectors = reddit_glove_vectors)
```

```
## INFO [2018-12-05 11:50:00] capital-common-countries: correct 191 out of 506, accuracy = 0.3775
## INFO [2018-12-05 11:50:22] capital-world: correct 419 out of 1594, accuracy = 0.2629
## INFO [2018-12-05 11:50:26] currency: correct 5 out of 376, accuracy = 0.0133
## INFO [2018-12-05 11:51:06] city-in-state: correct 1349 out of 2467, accuracy = 0.5468
## INFO [2018-12-05 11:51:12] family: correct 275 out of 506, accuracy = 0.5435
## INFO [2018-12-05 11:51:21] gram1-adjective-to-adverb: correct 108 out of 600, accuracy = 0.1800
## INFO [2018-12-05 11:51:28] gram2-opposite: correct 135 out of 552, accuracy = 0.2446
## INFO [2018-12-05 11:51:44] gram3-comparative: correct 820 out of 1122, accuracy = 0.7308
## INFO [2018-12-05 11:51:57] gram4-superlative: correct 375 out of 992, accuracy = 0.3780
## INFO [2018-12-05 11:52:08] gram5-present-participle: correct 538 out of 756, accuracy = 0.7116
## INFO [2018-12-05 11:52:30] gram6-nationality-adjective: correct 951 out of 1521, accuracy = 0.6252
## INFO [2018-12-05 11:52:46] gram7-past-tense: correct 588 out of 1122, accuracy = 0.5241
## INFO [2018-12-05 11:53:04] gram8-plural: correct 813 out of 1332, accuracy = 0.6104
## INFO [2018-12-05 11:53:12] gram9-plural-verbs: correct 272 out of 600, accuracy = 0.4533
## INFO [2018-12-05 11:53:12] OVERALL ACCURACY = 0.4869
```

Das Ergebnis von ~48% richtig beantworteten Aufgaben liegt in einem Ähnlichen Bereich, wie bei anderen vergleichbaren GloVe-Modellen (vgl. Lison & Kutuzov, 2017).

Datenbereinigung (II)

Im nächsten Schritt werden Autoren entfernt, die in mehr als einem der untersuchten Subreddits Kommentare verfasst haben, um Verfälschungen in der Auswertung zu vermeiden.

```
TD_authors <- reddit_df %>%
  filter (subreddit == "The_Donald") %>%
  distinct(author)
CON_authors <- reddit_df %>%
  filter (subreddit == "Conservative") %>%
  distinct(author)
NP_authors <- reddit_df %>%
  filter (subreddit == "NeutralPolitics") %>%
  distinct(author)
SFP_authors <- reddit_df %>%
  filter (subreddit == "SandersForPresident") %>%
  distinct(author)
LSC_authors <- reddit_df %>%
  filter (subreddit == "LateStageCapitalism") %>%
  distinct(author)

TD_CON <- intersect(TD_authors, CON_authors)
TD_NP <- intersect(TD_authors, NP_authors)
TD_SFP <- intersect(TD_authors, SFP_authors)
TD_LSC <- intersect(TD_authors, LSC_authors)
CON_NP <- intersect(CON_authors, NP_authors)
CON_SFP <- intersect(CON_authors, SFP_authors)
CON_LSC <- intersect(CON_authors, LSC_authors)
```

```

NP_SFP <- intersect(NP_authors, SFP_authors)
NP_LSC <- intersect(NP_authors, LSC_authors)
SFP_LSC <- intersect(SFP_authors, LSC_authors)

non_unique_authors <- TD_CON %>%
  union (TD_NP) %>%
  union (TD_SFP) %>%
  union (TD_LSC) %>%
  union (CON_NP) %>%
  union (CON_SFP) %>%
  union (CON_LSC) %>%
  union (NP_SFP) %>%
  union (NP_LSC) %>%
  union (SFP_LSC)

reddit_df <- anti_join(reddit_df, non_unique_authors, by = "author")

```

Nach diesem Schritt enthält der Datensatz noch 21,939,071 Kommentare.

Distributed Dictionary Representations (DDR)

Die Kodierung der Kommentare erfolgt mit der **Distributed Dictionary Representations (DDR)** Methode von Garten et al. (2018). Zuerst werden Konzeptrepräsentationen erstellt, daraufhin wird deren Kosinus-Ähnlichkeit mit den aggregierten Kommentarvektoren berechnet.

Konzeptrepräsentationen

Als Konzeptrepräsentationen werden die aggregierten Vektoren der “wichtigsten” Worte, sogenannter *seed words* aus dem *Moral Foundations Dictionary* (Graham et al., 2009) bezeichnet. Diese wurden vollständig aus der Studie von Garten et al. (2018) entnommen. Zur Berechnung dieser Vektoren wird die folgende Funktion definiert:

```

make_concept_rep <- function(model, seed_vec) {
  ## model: pretrained GloVe or word2vec word embeddings
  ## seed_vec: a vector of seed words for querying and aggregating

  concept_vec <- double(length = 100L)

  for (word in seed_vec){

    concept_vec <- concept_vec + as.double (model[word, , drop = FALSE])

  }
  return (concept_vec)
}

```

Daraufhin werden die *seed words* als Vektoren definiert und in die Funktion *make_concept_rep* eingesetzt:

```

seed_words_individualizing <- c("kindness", "compassion", "nurture", "empathy",
                                "suffer", "cruel", "hurt", "harm",
                                "fairness", "equality", "justice", "rights",
                                "cheat", "fraud", "unfair", "injustice")

seed_words_binding <- c("authority", "obey", "respect", "tradition",
                        "subversion", "disobey", "disrespect", "chaos",
                        "loyal", "solidarity", "patriot", "fidelity",

```

```

      "betray", "treason", "disloyal", "traitor",
      "purity", "sanctity", "sacred", "wholesome",
      "impurity", "depravity", "degradation", "unnatural")

individualizing_foundation <- make_concept_rep (model = reddit_glove_vectors,
                                              seed_vec = seed_words_individualizing)

binding_foundation <- make_concept_rep (model = reddit_glove_vectors,
                                       seed_vec = seed_words_binding)

```

Kodierung der Kommentare

Im ersten Schritt wird den Kommentaren eine (neue) individuelle ID zugewiesen.

```
reddit_df$comment_id <- 1:21939071
```

Daraufhin wird eine Iteratorfunktion definiert, um aus den Kommentaren iterativ in eine *Document-Term Matrix* herzustellen.

```

it <- itoken(iterable = reddit_df$body,
            ids = reddit_df$comment_id,
            preprocessor = tolower,
            tokenizer = word_tokenizer,
            n_chunks = 500)

```

Die kodierten Werte sollen ausschließlich auf den Worten basieren, die auch in dem zuvor berechneten GloVe-Modell vorkommen.

```

vectorizer <- vocab_vectorizer(pruned_vocab)

dtm = create_dtm(it, vectorizer)
dtm = normalize(dtm)

```

Um die so repräsentierten Kommentare im Vektorraum zu repräsentieren, werden die Wortvektoren über die normalisierten Kommentare summiert.

```
document_vecs = dtm %*% reddit_glove_vectors
```

In einem letzten Schritt kann nun die Kosinus-Ähnlichkeit zwischen Kommentaren und Konzeptrepräsentationen berechnet werden.

```

reddit_df$individualizing_foundation <- sim2(document_vecs, individualizing_foundation)
reddit_df$binding_foundation <- sim2(document_vecs, binding_foundation)

```

Kommentare, die keine einziges Wort enthalten, das im GloVe-Modell vorkommt, werden entfernt, da sie mit einer Kodierung von 0 das Ergebnis verzerren würden.

```

reddit_df <- reddit_df %>%
  filter (individualizing_foundation != 0 | binding_foundation != 0)

```

Datenbereinigung (III)

Da den abhängigen Variablen *individualizing_foundation* und *binding_foundation* keine intrinsische, bedeutsame Skalierung zueigen ist, können diese Variablen transformiert werden, um etwaige Probleme mit der Nicht-Normalität und Heteroskedastizität der Residuen vorzubeugen.

Für eine BoxCox-Transformation müssen alle Werte der Variablen positiv (und idealerweise über 1) sein, daher wird zu den Kosinus-Ähnlichkeiten eine Konstante addiert.

```
reddit_df <- reddit_df %>%
  mutate (author = factor(author),
          individualizing_foundation = individualizing_foundation + 2,
          binding_foundation = binding_foundation + 2)
```

Daraufhin werden die Werte für Lambda experimentell bestimmt. Für *individualizing_foundation* ergab sich ein Lambda von 9.4, für *binding_foundation* ein Lambda von 8.5.

```
ind_trans <- BoxCoxTrans(reddit_df$individualizing_foundation,
                        lambda = seq(0, 20, 1/10))

bind_trans <- BoxCoxTrans(reddit_df$binding_foundation,
                        lambda = seq(0, 20, 1/10))

reddit_df <- cbind(reddit_df,
                  trans_individualizing = predict(ind_trans,
                                                  reddit_df$individualizing_foundation))

reddit_df <- cbind(reddit_df,
                  trans_binding = predict(bind_trans,
                                          reddit_df$binding_foundation))
```

Z-Standardisierte Betas für längsschnittliche Untersuchungen bringen eine Reihe von Problemen mit sich (siehe Moeller, 2015). Daher empfehlen die Autoren eine *Proportion of Maximum Variance Skalierung* (POMS), auch *Min-Max Skalierung* genannt. Diese ermöglicht eine bessere Vergleichbarkeit der Koeffizienten und ist im Paket *BBmisc* verfügbar.

```
POMS = [(observed - minimum)/(maximum - minimum)]

reddit_df$individualizing <- normalize(reddit_df$trans_individualizing,
                                     method = "range", range = c(0, 1))

reddit_df$binding <- normalize(reddit_df$trans_binding,
                              method = "range", range = c(0, 1))
```

Um die unabhängige Variable *timediff* zu berechnen, wird für jeden individuellen Autoren der Zeitpunkt des Ersten Kommentars (im UTF-Format) vom Zeitpunkt jedes weiteren Posts abgezogen. Dieser Wert repräsentiert dann die Zeit, seit der ein Autor aktiv an der Konversation in einem Subreddit beteiligt ist.

```
min_by_authors <- reddit_df %>%
  group_by (author) %>%
  summarize (min_time = min(created_utc))

reddit_df <- reddit_df %>%
  left_join(min_by_authors) %>%
  mutate (timediff = created_utc - min_time)
```

Da die Einheit von *timediff* (Sekunden) zu klein ist, um bedeutungsvolle psychische Veränderungen feststellen zu können, wird der Wert jeweils durch 31,557,600 geteilt, so dass *timediff* in Jahren angegeben wird.

```
reddit_df$timediff <- reddit_df$timediff / 31557600
```

Für die Deskriptiven Statistiken wird darüber hinaus die Anzahl an Worten pro Kommentar gezählt.

```
reddit_corpus <- corpus(reddit_df, docid_field = "comment_id", text_field = "body")

reddit_df$wordno <- ntoken(reddit_corpus, remove_numbers = TRUE, remove_punct = TRUE)
```

Deskriptive Statistiken

```
reddit_df <- reddit_df %>%
  mutate (created = as_datetime(created_utc),
          author = factor(author),
          subreddit = factor(subreddit))%>%
  dplyr::select (author, timediff, created_utc, subreddit,
                individualizing, binding, wordno)
```

Überblick über den vollständigen Datensatz:

```
glimpse(reddit_df)
```

```
## Observations: 21,568,517
## Variables: 13
## $ body                <chr> "I bet Elon thinks Adam Smith was a...
## $ author              <fct> lilpoopybutt, fuckeverything2222, B...
## $ created_utc         <int> 1529122988, 1528921293, 1528866102,...
## $ subreddit           <fct> LateStageCapitalism, LateStageCapit...
## $ individualizing_foundation <dbl> 2.463201, 2.485693, 2.595141, 2.606...
## $ binding_foundation  <dbl> 2.301799, 2.354728, 2.373548, 2.428...
## $ wordno              <int> 11, 20, 80, 69, 18, 12, 6, 70, 22, ...
## $ trans_individualizing <dbl> 509.1910, 554.6189, 831.6303, 867.9...
## $ trans_binding       <dbl> 140.53713, 170.52080, 182.46713, 22...
## $ individualizing     <dbl> 0.25217687, 0.27478044, 0.41261304,...
## $ binding             <dbl> 0.18655942, 0.22725588, 0.24347050,...
## $ min_time            <dbl> 1528377780, 1516565511, 1504518976,...
## $ timediff            <dbl> 2.361422e-02, 3.915311e-01, 7.71513...
```

Die Deskriptiven Statistiken können auf zwei Ebenen getrennt werden - einerseits auf der Ebene der Autoren und andererseits auf der Ebene der Kommentare.

```
desc_df <- reddit_df %>%
  group_by (subreddit, author) %>%
  summarize (n_comments = n(),
            individualizing = mean(individualizing),
            binding = mean(binding))

descriptives_authors1 <- desc_df %>%
  summarise(
    N_authors = n_distinct(author, na.rm = TRUE),
    # number of comments
    max_comments = max(n_comments, na.rm = TRUE),
    min_comments = min(n_comments, na.rm = TRUE),
    mean_comments = mean(n_comments, na.rm = TRUE),
    sd_comments = sd(n_comments, na.rm = TRUE),
    median_comments = median(n_comments, na.rm = TRUE),
    # individualizing_foundation
    max_individ = max(individualizing, na.rm = TRUE),
    min_individ = min(individualizing, na.rm = TRUE),
    mean_individ = mean(individualizing, na.rm = TRUE),
    sd_individ = sd(individualizing, na.rm = TRUE),
    median_individ = median(individualizing, na.rm = TRUE),
    # binding_foundation
    max_binding = max(binding, na.rm = TRUE),
    min_binding = min(binding, na.rm = TRUE),
```



```

        mean_binding = mean(binding, na.rm = TRUE),
        sd_binding = sd(binding, na.rm = TRUE),
        median_binding = median(binding, na.rm = TRUE),
        # timespan
        max_timespan = max(timespan, na.rm = TRUE),
        min_timespan = min(timespan, na.rm = TRUE),
        mean_timespan = mean(timespan, na.rm = TRUE),
        sd_timespan = sd(timespan, na.rm = TRUE),
        median_timespan = median(timespan, na.rm = TRUE)
    )

descriptives_authors2 <- desc_df %>%
  group_by(subreddit) %>%
  summarise(
    N_authors = n_distinct(author, na.rm = TRUE),
    # number of comments
    max_comments = max(n_comments, na.rm = TRUE),
    min_comments = min(n_comments, na.rm = TRUE),
    sd_comments = sd(n_comments, na.rm = TRUE),
    mean_comments = mean(n_comments, na.rm = TRUE),
    median_comments = median(n_comments, na.rm = TRUE),
    # individualizing_foundation
    max_individ = max(individualizing, na.rm = TRUE),
    min_individ = min(individualizing, na.rm = TRUE),
    mean_individ = mean(individualizing, na.rm = TRUE),
    sd_individ = sd(individualizing, na.rm = TRUE),
    median_individ = median(individualizing, na.rm = TRUE),
    # binding_foundation
    max_binding = max(binding, na.rm = TRUE),
    min_binding = min(binding, na.rm = TRUE),
    mean_binding = mean(binding, na.rm = TRUE),
    sd_binding = sd(binding, na.rm = TRUE),
    median_binding = median(binding, na.rm = TRUE),
    # timespan
    max_timespan = max(timespan, na.rm = TRUE),
    min_timespan = min(timespan, na.rm = TRUE),
    mean_timespan = mean(timespan, na.rm = TRUE),
    sd_timespan = sd(timespan, na.rm = TRUE),
    median_timespan = median(timespan, na.rm = TRUE)
  )

```

Überblick über den innerhalb der Autoren gemittelten Datensatz:

```
glimpse(desc_df)
```

```

## Observations: 286,577
## Variables: 6
## $ subreddit      <fct> Conservative, Conservative, Conservative, Cons...
## $ author          <fct> --_o_--, --shaunoftheliving, --Solus, --Wh...
## $ n_comments      <int> 5, 6, 20, 16, 15, 7, 19, 4, 15, 4, 188, 9, 10,...
## $ individualizing <dbl> 0.4366499, 0.2536145, 0.3347979, 0.4736715, 0....
## $ binding         <dbl> 0.2933803, 0.2075318, 0.2568503, 0.3232007, 0....
## $ timespan        <dbl> 1.056760e-02, 5.462971e-01, 7.537887e-01, 6.66...

```

```
correlations <- cor(desc_df$individualizing, desc_df$binding)
```

```
correlations
```

```
## [1] 0.9036539
```

Für den gesamten Datensatz:

```
kable(descriptives_authors1[,1])
```

N_authors
286577

```
kable(descriptives_authors1[,2:6])
```

max_comments	min_comments	mean_comments	sd_comments	median_comments
50324	1	75.26255	377.9089	10

```
kable(descriptives_authors1[,7:11])
```

max_individ	min_individ	mean_individ	sd_individ	median_individ
0.7239977	0.001635	0.3404725	0.0895647	0.3424802

```
kable(descriptives_authors1[,12:16])
```

max_binding	min_binding	mean_binding	sd_binding	median_binding
0.6172891	0.0038112	0.2552928	0.0554911	0.2581257

```
kable(descriptives_authors1[,17:21])
```

max_timespan	min_timespan	mean_timespan	sd_timespan	median_timespan
8.448721	0	0.6101281	0.7042991	0.3767726

Aufgeteilt nach Subreddits:

```
kable(descriptives_authors2[,c(1, 2)])
```

subreddit	N_authors
Conservative	25212
LateStageCapitalism	27571
NeutralPolitics	6579
SandersForPresident	56581
The_Donald	170634

```
kable(descriptives_authors2[,c(1, 3:7)])
```

subreddit	max_comments	min_comments	sd_comments	mean_comments	median_comments
Conservative	10789	2	179.98231	32.54589	9
LateStageCapitalism	6839	1	77.05643	18.29582	7
NeutralPolitics	7839	2	141.88213	21.38668	8
SandersForPresident	12725	1	185.88826	42.68177	9
The_Donald	50324	1	468.84167	103.65962	12

```
kable(descriptives_authors2[,c(1, 8:12)])
```

subreddit	max_individ	min_individ	mean_individ	sd_individ	median_individ
Conservative	0.7050513	0.0350305	0.4102190	0.0682192	0.4119064
LateStageCapitalism	0.7239977	0.0534703	0.3873540	0.0734802	0.3887645
NeutralPolitics	0.6688226	0.0733037	0.4313059	0.0636701	0.4320929
SandersForPresident	0.7049986	0.0246985	0.3575607	0.0690726	0.3578045
The_Donald	0.6891003	0.0016350	0.3134235	0.0891839	0.3188273

```
kable(descriptives_authors2[,c(1, 13:17)])
```

subreddit	max_binding	min_binding	mean_binding	sd_binding	median_binding
Conservative	0.5442417	0.0454213	0.2916550	0.0441553	0.2911770
LateStageCapitalism	0.5152939	0.0561439	0.2672954	0.0433488	0.2677016
NeutralPolitics	0.4748664	0.0616484	0.2884360	0.0431886	0.2882130
SandersForPresident	0.4748359	0.0340292	0.2524644	0.0436069	0.2522988
The_Donald	0.6172891	0.0038112	0.2476408	0.0596260	0.2529876

```
kable(descriptives_authors2[,c(1, 18:22)])
```

subreddit	max_timespan	min_timespan	mean_timespan	sd_timespan	median_timespan
Conservative	8.448721	4.0e-07	0.8297996	1.2437171	0.2843106
LateStageCapitalism	2.827687	0.0e+00	0.5088599	0.5173418	0.3553901
NeutralPolitics	6.373922	9.9e-06	1.1767659	1.3867220	0.6441363
SandersForPresident	4.532740	0.0e+00	0.6079098	0.6935063	0.3495956
The_Donald	2.696772	0.0e+00	0.5729216	0.5523916	0.3983221

```
descriptives_comments1 <- reddit_df %>%
  summarize(
    max_created = max(created),
    min_created = min(created),
    max_wordno = max(wordno),
    min_wordno = min(wordno),
    mean_wordno = mean(wordno),
    median_wordno = median(wordno),
    sd_wordno = sd(wordno)
  )
```

```

descriptives_comments2 <- reddit_df %>%
  group_by(subreddit) %>%
  summarize(
    max_created = max(created),
    min_created = min(created),
    max_wordno = max(wordno),
    min_wordno = min(wordno),
    mean_wordno = mean(wordno),
    median_wordno = median(wordno),
    sd_wordno = sd(wordno)
  )

```

Für den gesamten Datensatz:

```
kable(descriptives_comments1[, 1:2])
```

max_created	min_created
2018-06-30 23:58:48	2008-08-15 21:45:04

```
kable(descriptives_comments1[, 3:7])
```

max_wordno	min_wordno	mean_wordno	median_wordno	sd_wordno
3334	0	26.62132	14	51.231

Aufgeteilt nach subreddits:

```
kable(descriptives_comments2[, c(1, 2:3)])
```

subreddit	max_created	min_created
Conservative	2018-06-30 23:58:48	2008-08-15 21:45:04
LateStageCapitalism	2018-06-30 23:51:07	2015-08-28 13:30:12
NeutralPolitics	2018-06-30 23:30:00	2012-02-14 04:36:52
SandersForPresident	2018-06-30 23:49:57	2013-12-05 23:20:50
The_Donald	2018-06-30 23:45:03	2015-07-14 05:12:01

```
kable(descriptives_comments2[, c(1, 4:8)])
```

subreddit	max_wordno	min_wordno	mean_wordno	median_wordno	sd_wordno
Conservative	2332	0	46.81783	26	68.03443
LateStageCapitalism	1803	0	40.12374	22	61.74373
NeutralPolitics	2415	0	92.29520	59	117.48596
SandersForPresident	2024	0	37.82265	19	60.70701
The_Donald	3334	0	23.24755	13	46.65926

Hypothesentestung

Für die Hypothesen 1a) und 1b) wird jeweils eine Mixed-Effects ANOVA gerechnet, für 2a) und 2b) jeweils der Übersichtlichkeit halber eine Mixed-Effects Regression pro Subreddit.

Es wird eine Optimierungsfunktion für die Auswertung mit *lme4* definiert, um die Berechnung zu beschleunigen.

```
nlopt <- function(par, fn, lower, upper, control) {  
  .nloptr <- res <- nloptr(par, fn, lb = lower, ub = upper,  
    opts = list(algorithm = "NLOPT_LN_BOBYQA", print_level = 1,  
      maxeval = 1000, xtol_abs = 1e-6, ftol_abs = 1e-6))  
  list(par = res$solution,  
    fval = res$objective,  
    conv = if (res$status > 0) 0 else res$status,  
    message = res$message  
  )  
}
```

Hypothese 1a)

Hypothese: Die moralischen Werte der Individualizing Foundation sind in den Gruppen des linken politischen Spektrums höher, als in denen des rechten politischen Spektrums.

Spezifizierung des Modells und der paarweisen Vergleiche:

```
hyp_1a <- lmer(individualizing ~ subreddit + (1 | author), data = reddit_df,  
  control = lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE))  
  
contr_h1a <- glht(hyp_1a, linfct=mcp(subreddit="Tukey"))  
  
summary_hyp_1a <- summary(hyp_1a)  
  
conf_h1a <- confint.merMod(hyp_1a, method = "Wald", level = 0.99)  
  
anova_h1a <- anova(hyp_1a)  
  
rsq_1a <- r2(hyp_1a)
```

Ergebnisse:

```
summary_hyp_1a  
  
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: individualizing ~ subreddit + (1 | author)  
## Data: reddit_df  
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)  
##  
## REML criterion at convergence: -26078671  
##  
## Scaled residuals:  
##      Min       1Q   Median       3Q      Max   
## -4.1010 -0.6688  0.0790  0.7051  5.2385   
##  
## Random effects:  
## Groups   Name                Variance Std.Dev.  
## author   (Intercept)  0.00345   0.05874  
## Residual                    0.01712  0.13084  
## Number of obs: 21568517, groups:  author, 286577  
##  
## Fixed effects:
```

```
##               Estimate Std. Error      df t value
## (Intercept)      4.103e-01  4.547e-04  2.540e+05  902.43
## subredditLateStageCapitalism -2.199e-02  6.389e-04  2.703e+05  -34.42
## subredditNeutralPolitics      2.238e-02  1.015e-03  2.721e+05   22.04
## subredditSandersForPresident -5.259e-02  5.451e-04  2.505e+05  -96.48
## subredditThe_Donald      -9.359e-02  4.851e-04  2.489e+05 -192.94
##               Pr(>|t|)
## (Intercept)      <2e-16 ***
## subredditLateStageCapitalism <2e-16 ***
## subredditNeutralPolitics      <2e-16 ***
## subredditSandersForPresident <2e-16 ***
## subredditThe_Donald      <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) sbrLSC sbrdNP sbrSFP
## sbrddtLtStC -0.712
## sbrddtNtrlP -0.448  0.319
## sbrddtSndFP -0.834  0.594  0.373
## sbrddtTh_Dn -0.937  0.667  0.420  0.782
```

Die Konfidenzintervalle der Effekte werden mit der Wald-Methode berechnet:

```
conf_h1a
```

```
##               0.5 %      99.5 %
## .sig01              NA          NA
## .sigma              NA          NA
## (Intercept)      0.40915500  0.41149740
## subredditLateStageCapitalism -0.02363661 -0.02034514
## subredditNeutralPolitics      0.01976222  0.02499313
## subredditSandersForPresident -0.05399844 -0.05119005
## subredditThe_Donald      -0.09484108 -0.09234207
```

Die Kennzahlen der ANOVA können so angezeigt werden:

```
anova_h1a
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##           Sum Sq Mean Sq NumDF DenDF F value    Pr(>F)
## subreddit 1168.8   292.2     4 259879  17068 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
rsq_1a
```

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: individualizing ~ subreddit + (1 | author)
##
##      Marginal R2: 0.029
##      Conditional R2: 0.192
```

Paarweise Vergleiche mit einem Tukey-Test:

```
summary(contr_h1a, test = adjusted("bonferroni"))

##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lmer(formula = individualizing ~ subreddit + (1 | author), data = reddit_df,
## control = lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE))
##
## Linear Hypotheses:
##
## Estimate Std. Error
## LateStageCapitalism - Conservative == 0 -0.0219909 0.0006389
## NeutralPolitics - Conservative == 0 0.0223777 0.0010154
## SandersForPresident - Conservative == 0 -0.0525942 0.0005451
## The_Donald - Conservative == 0 -0.0935916 0.0004851
## NeutralPolitics - LateStageCapitalism == 0 0.0443685 0.0010128
## SandersForPresident - LateStageCapitalism == 0 -0.0306034 0.0005403
## The_Donald - LateStageCapitalism == 0 -0.0716007 0.0004796
## SandersForPresident - NeutralPolitics == 0 -0.0749719 0.0009564
## The_Donald - NeutralPolitics == 0 -0.1159692 0.0009235
## The_Donald - SandersForPresident == 0 -0.0409973 0.0003450
##
## z value Pr(>|z|)
## LateStageCapitalism - Conservative == 0 -34.42 <2e-16 ***
## NeutralPolitics - Conservative == 0 22.04 <2e-16 ***
## SandersForPresident - Conservative == 0 -96.48 <2e-16 ***
## The_Donald - Conservative == 0 -192.94 <2e-16 ***
## NeutralPolitics - LateStageCapitalism == 0 43.81 <2e-16 ***
## SandersForPresident - LateStageCapitalism == 0 -56.64 <2e-16 ***
## The_Donald - LateStageCapitalism == 0 -149.28 <2e-16 ***
## SandersForPresident - NeutralPolitics == 0 -78.39 <2e-16 ***
## The_Donald - NeutralPolitics == 0 -125.58 <2e-16 ***
## The_Donald - SandersForPresident == 0 -118.84 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- bonferroni method)

confint(contr_h1a, level = 0.99)
```

```
##
## Simultaneous Confidence Intervals
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lmer(formula = individualizing ~ subreddit + (1 | author), data = reddit_df,
## control = lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE))
##
## Quantile = 3.2157
## 99% family-wise confidence level
##
##
## Linear Hypotheses:
```

	Estimate	lwr	upr
## LateStageCapitalism - Conservative == 0	-0.02199	-0.02405	-0.01994
## NeutralPolitics - Conservative == 0	0.02238	0.01911	0.02564
## SandersForPresident - Conservative == 0	-0.05259	-0.05435	-0.05084
## The_Donald - Conservative == 0	-0.09359	-0.09515	-0.09203
## NeutralPolitics - LateStageCapitalism == 0	0.04437	0.04111	0.04763
## SandersForPresident - LateStageCapitalism == 0	-0.03060	-0.03234	-0.02887
## The_Donald - LateStageCapitalism == 0	-0.07160	-0.07314	-0.07006
## SandersForPresident - NeutralPolitics == 0	-0.07497	-0.07805	-0.07190
## The_Donald - NeutralPolitics == 0	-0.11597	-0.11894	-0.11300
## The_Donald - SandersForPresident == 0	-0.04100	-0.04211	-0.03989

Hypothese 1b)

Hypothese: Die moralischen Werte der Binding Foundation sind in den Gruppen des rechten politischen Spektrums höher, als in denen des linken politischen Spektrums.

Spezifizierung des Modells und der paarweisen Vergleiche:

```
hyp_1b <- lmer (binding ~ subreddit + (1 | author), data = reddit_df,
               control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

contr_h1b <- glht(hyp_1b, linfct=mcp(subreddit="Tukey"))

summary_hyp_1b <- summary(hyp_1b)

conf_h1b <- confint.merMod(hyp_1b, method = "Wald", level = 0.99)

anova_h1b <- anova(hyp_1b)

rsq_1b <- r2(hyp_1b)
```

Ergebnisse:

```
summary_hyp_1b

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: binding ~ subreddit + (1 | author)
## Data: reddit_df
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -41810646
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.4021 -0.6338  0.0504  0.6484  8.3039
##
## Random effects:
##  Groups   Name                Variance Std.Dev.
##  author   (Intercept)  0.001272  0.03567
##  Residual                    0.008275  0.09097
## Number of obs: 21568517, groups:  author, 286577
##
## Fixed effects:
```



```
##               Estimate Std. Error      df    t value
## (Intercept)      2.918e-01  2.882e-04  2.479e+05 1012.611
## subredditLateStageCapitalism -2.371e-02  4.062e-04  2.673e+05  -58.369
## subredditNeutralPolitics      -2.670e-03  6.458e-04  2.699e+05   -4.135
## subredditSandersForPresident -3.936e-02  3.453e-04  2.436e+05 -114.009
## subredditThe_Donald      -4.083e-02  3.071e-04  2.417e+05 -132.940
##               Pr(>|t|)
## (Intercept)      < 2e-16 ***
## subredditLateStageCapitalism < 2e-16 ***
## subredditNeutralPolitics      3.55e-05 ***
## subredditSandersForPresident < 2e-16 ***
## subredditThe_Donald      < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr) sbrLSC sbrdNP sbrSFP
## sbrddtLtStC -0.709
## sbrddtNtrlP -0.446  0.317
## sbrddtSndFP -0.835  0.592  0.372
## sbrddtTh_Dn -0.938  0.666  0.419  0.783
```

Die Konfidenzintervalle der Effekte werden mit der Wald-Methode berechnet:

```
conf_h1b
```

```
##               0.5 %      99.5 %
## .sig01              NA          NA
## .sigma              NA          NA
## (Intercept)      0.291082209  0.292566866
## subredditLateStageCapitalism -0.024756787 -0.022664089
## subredditNeutralPolitics      -0.004333742 -0.001006769
## subredditSandersForPresident -0.040253536 -0.038474807
## subredditThe_Donald      -0.041621641 -0.040039392
```

Die Kennzahlen der ANOVA können so angezeigt werden:

```
anova_h1b
```

```
## Type III Analysis of Variance Table with Satterthwaite's method
##      Sum Sq Mean Sq NumDF  DenDF F value    Pr(>F)
## subreddit  190.1  47.524     4 254514  5743.2 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
rsq_1b
```

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ subreddit + (1 | author)
##
##      Marginal R2: 0.008
##      Conditional R2: 0.140
```

Paarweise Vergleiche mit einem Tukey-Test:

```
summary(contr_h1b, test = adjusted("bonferroni"))
```

```
##
## Simultaneous Tests for General Linear Hypotheses
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lmer(formula = binding ~ subreddit + (1 | author), data = reddit_df,
## control = lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE))
##
## Linear Hypotheses:
```

	Estimate	Std. Error	
## LateStageCapitalism - Conservative == 0	-0.0237104	0.0004062	
## NeutralPolitics - Conservative == 0	-0.0026703	0.0006458	
## SandersForPresident - Conservative == 0	-0.0393642	0.0003453	
## The_Donald - Conservative == 0	-0.0408305	0.0003071	
## NeutralPolitics - LateStageCapitalism == 0	0.0210402	0.0006450	
## SandersForPresident - LateStageCapitalism == 0	-0.0156537	0.0003437	
## The_Donald - LateStageCapitalism == 0	-0.0171201	0.0003053	
## SandersForPresident - NeutralPolitics == 0	-0.0366939	0.0006084	
## The_Donald - NeutralPolitics == 0	-0.0381603	0.0005876	
## The_Donald - SandersForPresident == 0	-0.0014663	0.0002178	
##	z value	Pr(> z)	
## LateStageCapitalism - Conservative == 0	-58.369	< 2e-16	***
## NeutralPolitics - Conservative == 0	-4.135	0.000355	***
## SandersForPresident - Conservative == 0	-114.009	< 2e-16	***
## The_Donald - Conservative == 0	-132.940	< 2e-16	***
## NeutralPolitics - LateStageCapitalism == 0	32.623	< 2e-16	***
## SandersForPresident - LateStageCapitalism == 0	-45.547	< 2e-16	***
## The_Donald - LateStageCapitalism == 0	-56.067	< 2e-16	***
## SandersForPresident - NeutralPolitics == 0	-60.310	< 2e-16	***
## The_Donald - NeutralPolitics == 0	-64.941	< 2e-16	***
## The_Donald - SandersForPresident == 0	-6.732	1.67e-10	***
## ---			
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1			
## (Adjusted p values reported -- bonferroni method)			

```
confint(contr_h1b, level = 0.99)
```

```
##
## Simultaneous Confidence Intervals
##
## Multiple Comparisons of Means: Tukey Contrasts
##
##
## Fit: lmer(formula = binding ~ subreddit + (1 | author), data = reddit_df,
## control = lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE))
##
## Quantile = 3.2151
## 99% family-wise confidence level
##
##
## Linear Hypotheses:
```

	Estimate	lwr
## LateStageCapitalism - Conservative == 0	-0.0237104	-0.0250165
## NeutralPolitics - Conservative == 0	-0.0026703	-0.0047466
## SandersForPresident - Conservative == 0	-0.0393642	-0.0404743
## The_Donald - Conservative == 0	-0.0408305	-0.0418180
## NeutralPolitics - LateStageCapitalism == 0	0.0210402	0.0189666
## SandersForPresident - LateStageCapitalism == 0	-0.0156537	-0.0167587
## The_Donald - LateStageCapitalism == 0	-0.0171201	-0.0181018
## SandersForPresident - NeutralPolitics == 0	-0.0366939	-0.0386501
## The_Donald - NeutralPolitics == 0	-0.0381603	-0.0400495
## The_Donald - SandersForPresident == 0	-0.0014663	-0.0021666
##	upr	
## LateStageCapitalism - Conservative == 0	-0.0224044	
## NeutralPolitics - Conservative == 0	-0.0005939	
## SandersForPresident - Conservative == 0	-0.0382541	
## The_Donald - Conservative == 0	-0.0398430	
## NeutralPolitics - LateStageCapitalism == 0	0.0231138	
## SandersForPresident - LateStageCapitalism == 0	-0.0145487	
## The_Donald - LateStageCapitalism == 0	-0.0161383	
## SandersForPresident - NeutralPolitics == 0	-0.0347378	
## The_Donald - NeutralPolitics == 0	-0.0362710	
## The_Donald - SandersForPresident == 0	-0.0007661	

Hypothese 2a)

Zur Überprüfung der zweiten Hypothese werden die Daten nach Subreddits aufgeteilt.

```
NP <- reddit_df %>% filter(subreddit == "NeutralPolitics")
TD <- reddit_df %>% filter(subreddit == "The_Donald")
CON <- reddit_df %>% filter(subreddit == "Conservative")
SFP <- reddit_df %>% filter(subreddit == "SandersForPresident")
LSC <- reddit_df %>% filter(subreddit == "LateStageCapitalism")
```

Hypothese: Die moralischen Werte der Individualizing Foundation werden in den Gruppen des linken politischen Spektrums über die Dauer der aktiven Mitgliedschaft höher, während sie in den Gruppen des rechten politischen Spektrums über die Zeit sinken.

Die Hypothese lässt sowohl lineare, als auch kurvilineare Effekte zu, daher werden für jedes Subreddit beide möglichen Modelle getestet und anhand des AIC und BIC verglichen.

```
NP_indi <- lmer (individualizing ~ timediff + (timediff | author), data = NP,
               control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

TD_indi <- lmer (individualizing ~ timediff + (timediff | author), data = TD,
               control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

CON_indi <- lmer (individualizing ~ timediff + (timediff | author), data = CON,
               control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

SFP_indi <- lmer (individualizing ~ timediff + (timediff | author), data = SFP,
               control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

LSC_indi <- lmer (individualizing ~ timediff + (timediff | author), data = LSC,
               control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

summary_NP_indi <- summary(NP_indi)
```

```

conf_NP_indi <- confint.merMod(NP_indi, method = "Wald", level = 0.99)

summary_TD_indi <- summary(TD_indi)
conf_TD_indi <- confint.merMod(TD_indi, method = "Wald", level = 0.99)

summary_CON_indi <- summary(CON_indi)
conf_CON_indi <- confint.merMod(CON_indi, method = "Wald", level = 0.99)

summary_SFP_indi <- summary(SFP_indi)
conf_SFP_indi <- confint.merMod(SFP_indi, method = "Wald", level = 0.99)

summary_LSC_indi <- summary(LSC_indi)
conf_LSC_indi <- confint.merMod(LSC_indi, method = "Wald", level = 0.99)

rsq_NP_indi <- r2(NP_indi)
rsq_TD_indi <- r2(TD_indi)
rsq_CON_indi <- r2(CON_indi)
rsq_SFP_indi <- r2(SFP_indi)
rsq_LSC_indi <- r2(LSC_indi)

NP_indi_poly <- lmer (individualizing ~ timediff + I(timediff^2) +(timediff + I(timediff^2)| author), d
                      control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

TD_indi_poly <- lmer (individualizing ~ timediff + I(timediff^2) +(timediff + I(timediff^2)| author), d
                      control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

CON_indi_poly <- lmer (individualizing ~ timediff + I(timediff^2) +(timediff + I(timediff^2)| author), d
                      control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

SFP_indi_poly <- lmer (individualizing ~ timediff + I(timediff^2) +(timediff + I(timediff^2)| author), d
                      control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

LSC_indi_poly <- lmer (individualizing ~ timediff + I(timediff^2) +(timediff + I(timediff^2)| author), d
                      control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

summary_NP_indi_poly <- summary(NP_indi_poly)
conf_NP_indi_poly <- confint.merMod(NP_indi_poly, method = "Wald", level = 0.99)

summary_TD_indi_poly <- summary(TD_indi_poly)
conf_TD_indi_poly <- confint.merMod(TD_indi_poly, method = "Wald", level = 0.99)

summary_CON_indi_poly <- summary(CON_indi_poly)
conf_CON_indi_poly <- confint.merMod(CON_indi_poly, method = "Wald", level = 0.99)

summary_SFP_indi_poly <- summary(SFP_indi_poly)
conf_SFP_indi_poly <- confint.merMod(SFP_indi_poly, method = "Wald", level = 0.99)

summary_LSC_indi_poly <- summary(LSC_indi_poly)
conf_LSC_indi_poly <- confint.merMod(LSC_indi_poly, method = "Wald", level = 0.99)

rsq_NP_indi_poly <- r2(NP_indi_poly)
rsq_TD_indi_poly <- r2(TD_indi_poly)
rsq_CON_indi_poly <- r2(CON_indi_poly)
rsq_SFP_indi_poly <- r2(SFP_indi_poly)

```

```
rsq_LSC_indi_poly <- r2(LSC_indi_poly)
```

```
comparison_NP_indi_models <- kable(anova(NP_indi, NP_indi_poly))
comparison_TD_indi_models <- kable(anova(TD_indi, TD_indi_poly))
comparison_CON_indi_models <- kable(anova(CON_indi, CON_indi_poly))
comparison_SFP_indi_models <- kable(anova(SFP_indi, SFP_indi_poly))
comparison_LSC_indi_models <- kable(anova(LSC_indi, LSC_indi_poly))
```

Ergebnisse in der Gruppe **Neutral_Politics**

```
summary_NP_indi
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: individualizing ~ timediff + (timediff | author)
## Data: NP
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -221831.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.4281 -0.5816  0.0564  0.6568  4.0348
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## author (Intercept) 0.0025310 0.05031
##        timediff    0.0002782 0.01668 -0.42
## Residual          0.0113777 0.10667
## Number of obs: 140703, groups: author, 6579
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  4.368e-01  8.137e-04 5.623e+03  536.84  <2e-16 ***
## timediff     -7.012e-03  5.936e-04 1.104e+03 -11.81  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## timediff -0.412
```

```
conf_NP_indi
```

```
##              0.5 %      99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sigma          NA          NA
## (Intercept)  0.434705177 0.43889685
## timediff     -0.008541451 -0.00548354
```

```
rsq_NP_indi
```

```
##
## R-Squared for Generalized Linear Mixed Model
```

```

##
## Family : gaussian (identity)
## Formula: individualizing ~ timediff + (timediff | author)
##
##      Marginal R2: 0.007
## Conditional R2: 0.194
summary_NP_indi_poly

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## individualizing ~ timediff + I(timediff^2) + (timediff + I(timediff^2) |
##      author)
##      Data: NP
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -222173.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.4532 -0.5806  0.0562  0.6540  4.0574
##
## Random effects:
##      Groups   Name                Variance Std.Dev. Corr
##      author   (Intercept)         0.0027284 0.05223
##               timediff            0.0023219 0.04819  -0.45
##               I(timediff^2)       0.0001062 0.01031   0.34 -0.97
## Residual                        0.0112942 0.10627
## Number of obs: 140703, groups:  author, 6579
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  4.364e-01  8.702e-04 5.577e+03 501.540 < 2e-16 ***
## timediff     -6.085e-03  1.474e-03 1.589e+03  -4.128 3.85e-05 ***
## I(timediff^2) -1.135e-04  3.793e-04 6.652e+02  -0.299   0.765
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) timdff
## timediff     -0.480
## I(timdff^2)   0.359 -0.930
conf_NP_indi_poly

##              0.5 %      99.5 %
## .sig01         NA         NA
## .sig02         NA         NA
## .sig03         NA         NA
## .sig04         NA         NA
## .sig05         NA         NA
## .sig06         NA         NA
## .sigma         NA         NA
## (Intercept)    0.434179661 0.4386624363
## timediff       -0.009881961 -0.0022882386

```

```
## I(timediff^2) -0.001090393 0.0008633819
```

```
rsq_NP_indi_poly
```

```
##
```

```
## R-Squared for Generalized Linear Mixed Model
```

```
##
```

```
## Family : gaussian (identity)
```

```
## Formula: individualizing ~ timediff + I(timediff^2) + (timediff + I(timediff^2) | author)
```

```
##
```

```
## Marginal R2: 0.006
```

```
## Conditional R2: 0.194
```

```
comparison_NP_indi_models
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
NP_indi	6	-221845.5	-221786.4	110928.8	-221857.5	NA	NA	NA
NP_indi_poly	10	-222193.2	-222094.7	111106.6	-222213.2	355.6665	4	0

Ergebnisse in der Gruppe **TheDonald**

```
summary_TD_indi
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
```

```
## lmerModLmerTest]
```

```
## Formula: individualizing ~ timediff + (timediff | author)
```

```
## Data: TD
```

```
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
```

```
##
```

```
## REML criterion at convergence: -21015554
```

```
##
```

```
## Scaled residuals:
```

```
## Min 1Q Median 3Q Max
```

```
## -3.5882 -0.6860 0.0742 0.7110 5.2266
```

```
##
```

```
## Random effects:
```

```
## Groups Name Variance Std.Dev. Corr
```

```
## author (Intercept) 0.004568 0.06759
```

```
## timediff 0.001496 0.03868 -0.56
```

```
## Residual 0.017503 0.13230
```

```
## Number of obs: 17687856, groups: author, 170634
```

```
##
```

```
## Fixed effects:
```

```
## Estimate Std. Error df t value Pr(>|t|)
```

```
## (Intercept) 3.160e-01 1.971e-04 1.268e+05 1603.069 < 2e-16 ***
```

```
## timediff 8.992e-04 2.313e-04 2.921e+04 3.888 0.000101 ***
```

```
## ---
```

```
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Correlation of Fixed Effects:
```

```
## (Intr)
```

```
## timediff -0.452
```

```
conf_TD_indi
```

```
## 0.5 % 99.5 %
```

```
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sigma          NA          NA
## (Intercept) 0.3154770826 0.316492538
## timediff    0.0003034736 0.001494995
```

```
rsq_TD_indi
```

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: individualizing ~ timediff + (timediff | author)
##
## Marginal R2: 0.000
## Conditional R2: 0.177
```

```
summary_TD_indi_poly
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## individualizing ~ timediff + I(timediff^2) + (timediff + I(timediff^2) |
## author)
## Data: TD
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -21031521
##
## Scaled residuals:
## Min      1Q  Median      3Q      Max
## -3.5531 -0.6856  0.0740  0.7107  5.1856
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## author (Intercept) 0.004912 0.07009
##          timediff 0.008942 0.09456 -0.56
##          I(timediff^2) 0.002485 0.04985 0.44 -0.96
## Residual 0.017469 0.13217
## Number of obs: 17687856, groups: author, 170634
##
## Fixed effects:
## Estimate Std. Error df t value Pr(>|t|)
## (Intercept) 3.172e-01 2.100e-04 1.235e+05 1510.37 <2e-16 ***
## timediff -1.082e-02 5.559e-04 3.516e+04 -19.46 <2e-16 ***
## I(timediff^2) 9.497e-03 3.626e-04 1.655e+04 26.20 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr) timdff
## timediff -0.508
## I(timdff^2) 0.377 -0.919
```



```
conf_TD_indi_poly
```

```
##              0.5 %      99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sig04          NA          NA
## .sig05          NA          NA
## .sig06          NA          NA
## .sigma          NA          NA
## (Intercept)    0.316671970 0.317753942
## timediff       -0.012250787 -0.009387217
## I(timediff^2)  0.008563554 0.010431307
```

```
rsq_TD_indi_poly
```

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: individualizing ~ timediff + I(timediff^2) + (timediff + I(timediff^2) | author)
##
##      Marginal R2: 0.000
##      Conditional R2: 0.174
```

```
comparison_TD_indi_models
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
TD_indi	6	-21015572	-21015484	10507792	-21015584	NA	NA	NA
TD_indi_poly	10	-21031546	-21031399	10515783	-21031566	15981.86	4	0

Ergebnisse in der Gruppe **Conservative**

```
summary_CON_indi
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: individualizing ~ timediff + (timediff | author)
##      Data: CON
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -1073486
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.3340 -0.5742  0.1003  0.6719  4.7390
##
## Random effects:
##  Groups   Name                Variance Std.Dev. Corr
##  author   (Intercept) 0.0025996 0.05099
##           timediff    0.0002297 0.01516  -0.43
## Residual                0.0152324 0.12342
## Number of obs: 820547, groups:  author, 25212
##
## Fixed effects:
```

```

##               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  4.124e-01  4.260e-04  2.066e+04  968.06  <2e-16 ***
## timediff    -5.545e-03  3.497e-04  2.158e+03  -15.86  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## timediff -0.375
conf_CON_indi

##               0.5 %      99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sigma          NA          NA
## (Intercept)  0.411273830  0.41346832
## timediff    -0.006445354 -0.00464402
rsq_CON_indi

##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: individualizing ~ timediff + (timediff | author)
##
##      Marginal R2: 0.003
##      Conditional R2: 0.147
summary_CON_indi_poly

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## individualizing ~ timediff + I(timediff^2) + (timediff + I(timediff^2) |
##      author)
##      Data: CON
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -1074310
##
## Scaled residuals:
##      Min      1Q  Median      3Q      Max
## -4.2830 -0.5734  0.1004  0.6708  4.7446
##
## Random effects:
##      Groups   Name                Variance Std.Dev. Corr
##      author   (Intercept)         2.726e-03 0.052207
##               timediff           1.732e-03 0.041613 -0.45
##               I(timediff^2)       6.345e-05 0.007966  0.36 -0.98
##      Residual                    1.519e-02 0.123240
## Number of obs: 820547, groups:  author, 25212
##
## Fixed effects:

```

```
##               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   4.132e-01  4.493e-04  2.036e+04  919.793   <2e-16 ***
## timediff      -1.135e-02  7.958e-04  3.654e+03 -14.264   <2e-16 ***
## I(timediff^2)  1.664e-03  1.910e-04  1.309e+03   8.713   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr) timdff
## timediff   -0.450
## I(timdff^2)  0.339 -0.914
```

conf_CON_indi_poly

```
##               0.5 %      99.5 %
## .sig01           NA         NA
## .sig02           NA         NA
## .sig03           NA         NA
## .sig04           NA         NA
## .sig05           NA         NA
## .sig06           NA         NA
## .sigma           NA         NA
## (Intercept)    0.412064804  0.414379216
## timediff       -0.013400740 -0.009301124
## I(timediff^2)  0.001172295  0.002156301
```

rsq_CON_indi_poly

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: individualizing ~ timediff + I(timediff^2) + (timediff + I(timediff^2) | author)
##
##      Marginal R2: 0.002
##      Conditional R2: 0.148
```

comparison_CON_indi_models

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
CON_indi	6	-1073501	-1073432	536756.7	-1073513	NA	NA	NA
CON_indi_poly	10	-1074333	-1074217	537176.7	-1074353	839.9145	4	0

Ergebnisse in der Gruppe **SandersForPresident**

summary_SFP_indi

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: individualizing ~ timediff + (timediff | author)
##      Data: SFP
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -3318415
##
## Scaled residuals:
```

```
##      Min      1Q  Median      3Q      Max
## -5.2579 -0.5932  0.1009   0.6880  4.8392
##
## Random effects:
##   Groups   Name                Variance Std.Dev. Corr
##   author   (Intercept)  0.003212  0.05668
##             timediff     0.002616  0.05114  -0.48
##   Residual                    0.014269  0.11945
## Number of obs: 2414977, groups:  author, 56581
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept) 3.554e-01  3.066e-04 4.893e+04 1158.96  <2e-16 ***
## timediff     5.887e-03  5.175e-04 1.382e+04   11.38  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## timediff -0.464
```

conf_SFP_indi

```
##              0.5 %      99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sigma          NA          NA
## (Intercept) 0.354561197 0.356140760
## timediff     0.004553973 0.007220025
```

rsq_SFP_indi

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: individualizing ~ timediff + (timediff | author)
##
##      Marginal R2: 0.000
##      Conditional R2: 0.175
```

summary_SFP_indi_poly

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula:
## individualizing ~ timediff + I(timediff^2) + (timediff + I(timediff^2) |
##      author)
##      Data: SFP
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -3324004
##
## Scaled residuals:
##      Min      1Q  Median      3Q      Max
```

```

## -4.3022 -0.5927 0.1000 0.6871 4.8410
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## author (Intercept) 0.003439 0.05864
## timediff 0.012292 0.11087 -0.50
## I(timediff^2) 0.002081 0.04562 0.38 -0.97
## Residual 0.014212 0.11921
## Number of obs: 2414977, groups: author, 56581
##
## Fixed effects:
## Estimate Std. Error df t value Pr(>|t|)
## (Intercept) 3.571e-01 3.288e-04 4.809e+04 1086.077 <2e-16 ***
## timediff -9.573e-03 1.057e-03 1.661e+04 -9.053 <2e-16 ***
## I(timediff^2) 1.222e-02 5.417e-04 5.583e+03 22.556 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr) timdff
## timediff -0.533
## I(timdff^2) 0.390 -0.907

```

conf_SFP_indi_poly

```

## 0.5 % 99.5 %
## .sig01 NA NA
## .sig02 NA NA
## .sig03 NA NA
## .sig04 NA NA
## .sig05 NA NA
## .sig06 NA NA
## .sigma NA NA
## (Intercept) 0.35620563 0.357899258
## timediff -0.01229626 -0.006849144
## I(timediff^2) 0.01082298 0.013613599

```

rsq_SFP_indi_poly

```

##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: individualizing ~ timediff + I(timediff^2) + (timediff + I(timediff^2) | author)
##
## Marginal R2: 0.003
## Conditional R2: 0.175

```

comparison_SFP_indi_models

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
SFP_indi	6	-3318431	-3318355	1659221	-3318443	NA	NA	NA
SFP_indi_poly	10	-3324026	-3323899	1662023	-3324046	5602.618	4	0

Ergebnisse in der Gruppe **LateStageCapitalism**

summary_LSC_indi

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: individualizing ~ timediff + (timediff | author)
## Data: LSC
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -561313.9
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.6147 -0.5919  0.0965  0.6816  4.2919
##
## Random effects:
## Groups   Name                Variance Std.Dev. Corr
## author   (Intercept)  0.002951  0.05432
##          timediff      0.001348  0.03671  -0.48
## Residual                    0.018223  0.13499
## Number of obs: 504434, groups:  author, 27571
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  3.893e-01  4.754e-04  2.206e+04 818.965 < 2e-16 ***
## timediff     -4.585e-03  7.998e-04  3.880e+03 -5.732 1.07e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## timediff -0.489
```

conf_LSC_indi

```
##              0.5 %      99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sigma          NA          NA
## (Intercept)  0.38811472  0.390563841
## timediff     -0.00664483 -0.002524493
```

rsq_LSC_indi

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: individualizing ~ timediff + (timediff | author)
##
## Marginal R2: 0.000
## Conditional R2: 0.128
```

summary_LSC_indi_poly

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
```

```

## lmerModLmerTest]
## Formula:
## individualizing ~ timediff + I(timediff^2) + (timediff + I(timediff^2) |
##   author)
##   Data: LSC
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -561629
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.6319 -0.5908  0.0963  0.6805  4.3586
##
## Random effects:
##   Groups   Name                Variance Std.Dev. Corr
##   author   (Intercept)         0.003092 0.05561
##            timediff            0.007882 0.08878  -0.45
##            I(timediff^2)       0.001635 0.04043   0.32 -0.96
## Residual                   0.018168 0.13479
## Number of obs: 504434, groups:  author, 27571
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   3.901e-01  5.109e-04 2.120e+04 763.708 < 2e-16 ***
## timediff      -1.400e-02  1.817e-03 2.598e+03 -7.702 1.89e-14 ***
## I(timediff^2)  7.179e-03  1.174e-03 7.103e+02  6.115 1.59e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) timdff
## timediff      -0.535
## I(timdff^2)    0.380 -0.905

```

```

conf_LSC_indi_poly

##              0.5 %       99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sig04          NA          NA
## .sig05          NA          NA
## .sig06          NA          NA
## .sigma          NA          NA
## (Intercept)    0.388829399 0.391461158
## timediff       -0.018678891 -0.009316091
## I(timediff^2)  0.004155207 0.010203057

```

```

rsq_LSC_indi_poly

##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: individualizing ~ timediff + I(timediff^2) + (timediff + I(timediff^2) | author)
##

```

```
##      Marginal R2: 0.000
## Conditional R2: 0.129
comparison_LSC_indi_models
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
LSC_indi	6	-561328.1	-561261.3	280670.0	-561340.1	NA	NA	NA
LSC_indi_poly	10	-561646.9	-561535.6	280833.4	-561666.9	326.8158	4	0

Hypothese 2b)

Hypothese: Die moralischen Werte der Binding Foundation werden in den Gruppen des rechten politischen Spektrums über die Dauer der aktiven Mitgliedschaft höher, während sie in den Gruppen des linken politischen Spektrums über die Zeit sinken. Auch hier wird sowohl für lineare, als auch für polynome Effekte getestet.

```
NP_bindi <- lmer (binding ~ timediff + (timediff | author), data = NP,
                  control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

TD_bindi <- lmer (binding ~ timediff + (timediff | author), data = TD,
                  control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

CON_bindi <- lmer (binding ~ timediff + (timediff | author), data = CON,
                  control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

SFP_bindi <- lmer (binding ~ timediff + (timediff | author), data = SFP,
                  control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

LSC_bindi <- lmer (binding ~ timediff + (timediff | author), data = LSC,
                  control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

summary_NP_bindi <- summary(NP_bindi)
conf_NP_bindi <- confint.merMod(NP_bindi, method = "Wald", level = 0.99)

summary_TD_bindi <- summary(TD_bindi)
conf_TD_bindi <- confint.merMod(TD_bindi, method = "Wald", level = 0.99)

summary_CON_bindi <- summary(CON_bindi)
conf_CON_bindi <- confint.merMod(CON_bindi, method = "Wald", level = 0.99)

summary_SFP_bindi <- summary(SFP_bindi)
conf_SFP_bindi <- confint.merMod(SFP_bindi, method = "Wald", level = 0.99)

summary_LSC_bindi <- summary(LSC_bindi)
conf_LSC_bindi <- confint.merMod(LSC_bindi, method = "Wald", level = 0.99)

rsq_NP_bindi <- r2(NP_bindi)
rsq_TD_bindi <- r2(TD_bindi)
rsq_CON_bindi <- r2(CON_bindi)
rsq_SFP_bindi <- r2(SFP_bindi)
rsq_LSC_bindi <- r2(LSC_bindi)

NP_bindi_poly <- lmer (binding ~ timediff + (timediff + I(timediff^2) | author), data = NP,
                      control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))
```



```

TD_bindi_poly <- lmer (binding ~ timediff + (timediff + I(timediff^2) | author), data = TD,
  control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

CON_bindi_poly <- lmer (binding ~ timediff + (timediff + I(timediff^2) | author), data = CON,
  control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

SFP_bindi_poly <- lmer (binding ~ timediff + (timediff + I(timediff^2) | author), data = SFP,
  control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

LSC_bindi_poly <- lmer (binding ~ timediff + (timediff + I(timediff^2) | author), data = LSC,
  control = lmerControl (optimizer = "nloptwrap", calc.derivs = FALSE))

summary_NP_bindi_poly <- summary(NP_bindi_poly)
conf_NP_bindi_poly <- confint.merMod(NP_bindi_poly, method = "Wald", level = 0.99)

summary_TD_bindi_poly <- summary(TD_bindi_poly)
conf_TD_bindi_poly <- confint.merMod(TD_bindi_poly, method = "Wald", level = 0.99)

summary_CON_bindi_poly <- summary(CON_bindi_poly)
conf_CON_bindi_poly <- confint.merMod(CON_bindi_poly, method = "Wald", level = 0.99)

summary_SFP_bindi_poly <- summary(SFP_bindi_poly)
conf_SFP_bindi_poly <- confint.merMod(SFP_bindi_poly, method = "Wald", level = 0.99)

summary_LSC_bindi_poly <- summary(LSC_bindi_poly)
conf_LSC_bindi_poly <- confint.merMod(LSC_bindi_poly, method = "Wald", level = 0.99)

rsq_NP_bindi_poly <- r2(NP_bindi_poly)
rsq_TD_bindi_poly <- r2(TD_bindi_poly)
rsq_CON_bindi_poly <- r2(CON_bindi_poly)
rsq_SFP_bindi_poly <- r2(SFP_bindi_poly)
rsq_LSC_bindi_poly <- r2(LSC_bindi_poly)

comparison_NP_bindi_models <- kable(anova(NP_bindi, NP_bindi_poly))
comparison_TD_bindi_models <- kable(anova(TD_bindi, TD_bindi_poly))
comparison_CON_bindi_models <- kable(anova(CON_bindi, CON_bindi_poly))
comparison_SFP_bindi_models <- kable(anova(SFP_bindi, SFP_bindi_poly))
comparison_LSC_bindi_models <- kable(anova(LSC_bindi, LSC_bindi_poly))

```

Ergebnisse in der Gruppe **Neutral_Politics**

```
summary_NP_bindi
```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: binding ~ timediff + (timediff | author)
## Data: NP
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -330060.7
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.1476 -0.6065 -0.0178  0.6227  6.1241
##

```

```

## Random effects:
##   Groups   Name                Variance Std.Dev. Corr
##   author   (Intercept) 0.0011650 0.03413
##           timediff    0.0001402 0.01184  -0.47
##   Residual                0.0052747 0.07263
## Number of obs: 140703, groups:  author, 6579
##
## Fixed effects:
##               Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  2.910e-01  5.531e-04  5.554e+03 526.011 < 2e-16 ***
## timediff     -3.249e-03  4.070e-04  1.028e+03  -7.985 3.75e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##           (Intr)
## timediff -0.430

```

```

conf_NP_bindi

```

```

##               0.5 %           99.5 %
## .sig01           NA           NA
## .sig02           NA           NA
## .sig03           NA           NA
## .sigma           NA           NA
## (Intercept)  0.28952850  0.292378042
## timediff     -0.00429756 -0.002201087

```

```

rsq_NP_bindi

```

```

##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ timediff + (timediff | author)
##
##   Marginal R2: 0.003
##   Conditional R2: 0.187

```

```

summary_NP_bindi_poly

```

```

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: binding ~ timediff + I(timediff^2) + (timediff + I(timediff^2) |
##   author)
##   Data: NP
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -330510.5
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.1901 -0.6066 -0.0155  0.6177  5.9868
##
## Random effects:
##   Groups   Name                Variance Std.Dev. Corr

```

```
## author (Intercept) 1.277e-03 0.035731
##          timediff 1.479e-03 0.038453 -0.49
##          I(timediff^2) 7.631e-05 0.008736 0.38 -0.98
## Residual 5.222e-03 0.072262
## Number of obs: 140703, groups: author, 6579
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  2.909e-01  5.955e-04  5.541e+03 488.515 < 2e-16 ***
## timediff     -3.301e-03  1.073e-03  1.662e+03  -3.076  0.00213 **
## I(timediff^2)  8.517e-05  2.827e-04  7.321e+02   0.301  0.76328
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) timdff
## timediff     -0.497
## I(timdff^2)  0.378 -0.938
```

conf_NP_bindi_poly

```
##              0.5 %      99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sig04          NA          NA
## .sig05          NA          NA
## .sig06          NA          NA
## .sigma          NA          NA
## (Intercept)  0.289377742  0.2924455655
## timediff     -0.006064702 -0.0005367915
## I(timediff^2) -0.000643007  0.0008133553
```

rsq_NP_bindi_poly

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ timediff + I(timediff^2) + (timediff + I(timediff^2) | author)
##
##      Marginal R2: 0.003
##      Conditional R2: 0.200
```

comparison_NP_bindi_models

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
NP_bindi	6	-330075.8	-330016.7	165043.9	-330087.8	NA	NA	NA
NP_bindi_poly	10	-330532.3	-330433.8	165276.2	-330552.3	464.4803	4	0

Ergebnisse in der Gruppe **TheDonald**

summary_TD_bindi

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
```

```
## Formula: binding ~ timediff + (timediff | author)
## Data: TD
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -33610659
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.6978 -0.6439  0.0495  0.6515  8.1564
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## author (Intercept) 0.0017860 0.04226
##        timediff    0.0006255 0.02501 -0.61
## Residual          0.0086041 0.09276
## Number of obs: 17687856, groups: author, 170634
##
## Fixed effects:
## Estimate Std. Error df t value Pr(>|t|)
## (Intercept) 2.505e-01 1.272e-04 1.184e+05 1969.786 < 2e-16 ***
## timediff    7.656e-04 1.524e-04 2.765e+04 5.024 5.09e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## timediff -0.482
```

```
conf_TD_bindi
```

```
##           0.5 %      99.5 %
## .sig01      NA      NA
## .sig02      NA      NA
## .sig03      NA      NA
## .sigma      NA      NA
## (Intercept) 0.2501486033 0.250803683
## timediff    0.0003730961 0.001158175
```

```
rsq_TD_bindi
```

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ timediff + (timediff | author)
##
## Marginal R2: 0.000
## Conditional R2: 0.142
```

```
summary_TD_bindi_poly
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: binding ~ timediff + I(timediff^2) + (timediff + I(timediff^2) |
## author)
## Data: TD
```

```

## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -33622880
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.7102 -0.6438  0.0494  0.6512  8.1703
##
## Random effects:
##   Groups   Name                Variance Std.Dev. Corr
##   author   (Intercept)         0.001949 0.04415
##            timediff            0.003819 0.06180 -0.60
##            I(timediff^2)        0.001027 0.03205  0.48 -0.96
## Residual                    0.008590 0.09268
## Number of obs: 17687856, groups:  author, 170634
##
## Fixed effects:
##              Estimate Std. Error      df    t value Pr(>|t|)
## (Intercept)   2.507e-01  1.367e-04  1.147e+05 1833.843 < 2e-16 ***
## timediff      -1.824e-03  3.704e-04  3.408e+04  -4.925 8.48e-07 ***
## I(timediff^2)  2.213e-03  2.390e-04  1.579e+04   9.258 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) timdff
## timediff      -0.537
## I(timdff^2)    0.400 -0.922

```

```

conf_TD_bindi_poly

```

```

##              0.5 %          99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sig04          NA          NA
## .sig05          NA          NA
## .sig06          NA          NA
## .sigma          NA          NA
## (Intercept)    0.250334523  0.2510387551
## timediff       -0.002778602 -0.0008702109
## I(timediff^2)  0.001597146  0.0028285215

```

```

rsq_TD_bindi_poly

```

```

##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ timediff + I(timediff^2) + (timediff + I(timediff^2) | author)
##
##      Marginal R2: 0.000
##      Conditional R2: 0.139

```

```

comparison_TD_bindi_models

```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
TD_bindi	6	-33610679	-33610591	16805345	-33610691	NA	NA	NA
TD_bindi_poly	10	-33622907	-33622760	16811464	-33622927	12236.41	4	0

Ergebnisse in der Gruppe **Conservative**

summary_CON_bindi

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: binding ~ timediff + (timediff | author)
## Data: CON
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -1685301
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -4.0138 -0.5867  0.0414  0.6175  7.4446
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## author (Intercept) 9.692e-04 0.031131
##        timediff    9.975e-05 0.009987 -0.48
## Residual          7.262e-03 0.085219
## Number of obs: 820547, groups: author, 25212
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)  2.927e-01  2.730e-04 1.961e+04 1072.30 <2e-16 ***
## timediff     -2.349e-03  2.308e-04  2.126e+03 -10.18 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr)
## timediff -0.400
```

conf_CON_bindi

```
##              0.5 %      99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sigma          NA          NA
## (Intercept)  0.292006034  0.293412305
## timediff     -0.002943827 -0.001755008
```

rsq_CON_bindi

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ timediff + (timediff | author)
```

```
##
##   Marginal R2: 0.001
##   Conditional R2: 0.116
summary_CON_bindi_poly

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: binding ~ timediff + I(timediff^2) + (timediff + I(timediff^2) |
##   author)
##   Data: CON
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -1686100
##
## Scaled residuals:
##   Min       1Q   Median       3Q      Max
## -4.0820 -0.5862  0.0411  0.6169  7.3959
##
## Random effects:
##   Groups   Name                Variance Std.Dev. Corr
##   author   (Intercept)         1.035e-03 0.03217
##            timediff            8.024e-04 0.02833  -0.50
##            I(timediff^2)       2.863e-05 0.00535   0.40 -0.98
## Residual                        7.241e-03 0.08509
## Number of obs: 820547, groups:  author, 25212
##
## Fixed effects:
##              Estimate Std. Error      df    t value Pr(>|t|)
## (Intercept)   2.931e-01  2.905e-04 1.948e+04 1009.018 < 2e-16 ***
## timediff      -4.954e-03  5.317e-04  3.674e+03  -9.319 < 2e-16 ***
## I(timediff^2)  7.487e-04  1.260e-04  1.282e+03   5.942 3.61e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr) timdff
## timediff      -0.477
## I(timdff^2)    0.361 -0.918
```

```
conf_CON_bindi_poly

##              0.5 %      99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sig04          NA          NA
## .sig05          NA          NA
## .sig06          NA          NA
## .sigma          NA          NA
## (Intercept)    0.2923368000 0.293833179
## timediff       -0.0063238505 -0.003584932
## I(timediff^2)  0.0004241389 0.001073173
```

```
rsq_CON_bindi_poly
```

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ timediff + I(timediff^2) + (timediff + I(timediff^2) | author)
##
## Marginal R2: 0.001
## Conditional R2: 0.118
```

```
comparison_CON_bindi_models
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
CON_bindi	6	-1685318	-1685249	842665.2	-1685330	NA	NA	NA
CON_bindi_poly	10	-1686126	-1686010	843073.1	-1686146	815.7251	4	0

Ergebnisse in der Gruppe **SandersForPresident**

```
summary_SFP_bindi
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: binding ~ timediff + (timediff | author)
## Data: SFP
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -5312119
##
## Scaled residuals:
## Min 1Q Median 3Q Max
## -5.0011 -0.6043 0.0622 0.6550 7.4761
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## author (Intercept) 0.001204 0.03469
## timediff 0.001091 0.03303 -0.48
## Residual 0.006266 0.07916
## Number of obs: 2414977, groups: author, 56581
##
## Fixed effects:
## Estimate Std. Error df t value Pr(>|t|)
## (Intercept) 2.501e-01 1.930e-04 4.799e+04 1295.39 <2e-16 ***
## timediff 7.965e-03 3.368e-04 1.345e+04 23.64 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr)
## timediff -0.475
```

```
conf_SFP_bindi
```

```
## 0.5 % 99.5 %
## .sig01 NA NA
```



```
## .sig02          NA          NA
## .sig03          NA          NA
## .sigma          NA          NA
## (Intercept) 0.249557700 0.250552147
## timediff     0.007097071 0.008832374
```

```
rsq_SFP_bindi
```

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ timediff + (timediff | author)
##
## Marginal R2: 0.002
## Conditional R2: 0.155
```

```
summary_SFP_bindi_poly
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: binding ~ timediff + I(timediff^2) + (timediff + I(timediff^2) |
## author)
## Data: SFP
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -5317780
##
## Scaled residuals:
## Min      1Q  Median      3Q      Max
## -3.9615 -0.6039  0.0620  0.6539  7.4925
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## author (Intercept) 0.001291 0.03594
##          timediff 0.005607 0.07488 -0.50
##          I(timediff^2) 0.001008 0.03175 0.36 -0.97
## Residual 0.006240 0.07899
## Number of obs: 2414977, groups: author, 56581
##
## Fixed effects:
## Estimate Std. Error df t value Pr(>|t|)
## (Intercept) 2.506e-01 2.079e-04 4.712e+04 1205.280 < 2e-16 ***
## timediff 2.651e-03 7.043e-04 1.703e+04 3.764 0.000168 ***
## I(timediff^2) 4.665e-03 3.648e-04 5.811e+03 12.786 < 2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
## (Intr) timdff
## timediff -0.543
## I(timdff^2) 0.397 -0.911
```

```
conf_SFP_bindi_poly
```

```
##          0.5 %      99.5 %
```

```
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sig04          NA          NA
## .sig05          NA          NA
## .sig06          NA          NA
## .sigma          NA          NA
## (Intercept)    0.2501017135 0.251172998
## timediff       0.0008364773 0.004464759
## I(timediff^2)  0.0037251384 0.005604629
```

```
rsq_SFP_bindi_poly
```

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ timediff + I(timediff^2) + (timediff + I(timediff^2) | author)
##
## Marginal R2: 0.003
## Conditional R2: 0.156
```

```
comparison_SFP_bindi_models
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
SFP_bindi	6	-5312137	-5312061	2656075	-5312149	NA	NA	NA
SFP_bindi_poly	10	-5317804	-5317677	2658912	-5317824	5675.354	4	0

Ergebnisse in der Gruppe **LateStageCapitalism**

```
summary_LSC_bindi
```

```
## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: binding ~ timediff + (timediff | author)
## Data: LSC
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -1039589
##
## Scaled residuals:
## Min      1Q  Median      3Q      Max
## -3.6792 -0.5811  0.0570  0.6330  6.9379
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## author (Intercept) 0.0008876 0.02979
##          timediff  0.0004834 0.02199 -0.50
## Residual          0.0071181 0.08437
## Number of obs: 504434, groups: author, 27571
##
## Fixed effects:
## Estimate Std. Error df t value Pr(>|t|)
## (Intercept) 2.685e-01 2.781e-04 2.077e+04 965.655 < 2e-16 ***
## timediff -1.647e-03 4.851e-04 3.818e+03 -3.395 0.000693 ***
```

```

## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##      (Intr)
## timediff -0.512
conf_LSC_bindi

##              0.5 %      99.5 %
## .sig01          NA          NA
## .sig02          NA          NA
## .sig03          NA          NA
## .sigma          NA          NA
## (Intercept)  0.267832030  0.2692647038
## timediff    -0.002896863 -0.0003975494
rsq_LSC_bindi

##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ timediff + (timediff | author)
##
##      Marginal R2: 0.000
##      Conditional R2: 0.101
summary_LSC_bindi_poly

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [
## lmerModLmerTest]
## Formula: binding ~ timediff + I(timediff^2) + (timediff + I(timediff^2) |
##      author)
##      Data: LSC
## Control: lmerControl(optimizer = "nloptwrap", calc.derivs = FALSE)
##
## REML criterion at convergence: -1039823
##
## Scaled residuals:
##      Min      1Q  Median      3Q      Max
## -3.7297 -0.5806  0.0567  0.6321  6.9530
##
## Random effects:
##      Groups   Name      Variance Std.Dev. Corr
##      author   (Intercept)  0.0009409  0.03067
##              timediff     0.0026319  0.05130  -0.48
##              I(timediff^2) 0.0004828  0.02197   0.35 -0.96
##      Residual              0.0070998  0.08426
## Number of obs: 504434, groups:  author, 27571
##
## Fixed effects:
##              Estimate Std. Error      df t value Pr(>|t|)
## (Intercept)   2.689e-01  3.007e-04 2.006e+04 894.032 < 2e-16 ***
## timediff      -5.154e-03  1.083e-03  2.464e+03  -4.760 2.04e-06 ***
## I(timediff^2)  2.676e-03  6.881e-04  6.405e+02   3.889 0.000111 ***

```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) timediff
## timediff   -0.557
## I(timediff^2)  0.394 -0.902
```

```
conf_LSC_bindi_poly
```

```
##          0.5 %      99.5 %
## .sig01          NA      NA
## .sig02          NA      NA
## .sig03          NA      NA
## .sig04          NA      NA
## .sig05          NA      NA
## .sig06          NA      NA
## .sigma          NA      NA
## (Intercept)  0.2680772098 0.269626408
## timediff     -0.0079427910 -0.002365175
## I(timediff^2) 0.0009036721 0.004448598
```

```
rsq_LSC_bindi_poly
```

```
##
## R-Squared for Generalized Linear Mixed Model
##
## Family : gaussian (identity)
## Formula: binding ~ timediff + I(timediff^2) + (timediff + I(timediff^2) | author)
##
##      Marginal R2: 0.000
##      Conditional R2: 0.102
```

```
comparison_LSC_bindi_models
```

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
LSC_bindi	6	-1039606	-1039539	519808.8	-1039618	NA	NA	NA
LSC_bindi_poly	10	-1039844	-1039733	519932.0	-1039864	246.3645	4	0