

TOPIC

Dynamische und reproduzierbare Berichte mit R

Daniela Palleschi

2023-07-03

Inhaltsverzeichnis

Aufgabenübersicht	2
Beabsichtigte Lernergebnisse	2
Kenntnisse und Fähigkeiten	2
Kommunikation	2
Bewertung	2
Tipps	3
Biondo, Soilemezidi, und Mancini (2022): Verarbeitung der Kongruenz der Zeitformen der Verben	3
Variablen	4
Vorhersagen	4
1 Einrichten	5
1.1 YAML	5
1.2 Struktur des Dokuments	5
1.3 Pakete	6
1.4 Datenbeschreibung	6
2 Data wrangling	7
2.1 Data tidying	8
3 Plot-Interpretation	8
3.1 Distributions	9
3.2 Zusammenfassende Statistiken	11
4 Data summaries	13
5 Plotten der Daten	14
5.1 Verteilungen	14
5.2 Split-violin plots	15

6 Code-Chunk-Optionen	17
7 Session Info einbeziehen	17
8 Bericht einreichen	18
Session Info	18

```
knitr::opts_chunk$set(eval = T, # evaluate chunks
  echo = T, # 'print code chunk?'
  message = F, # 'print messages (e.g., warnings)?'
  error = F, # stop when error encountered
  warning = F) # don't print warnings
```

Aufgabenübersicht

Für diesen Bericht werden Sie Diagramme interpretieren, Daten einlesen und verarbeiten und 2 Tabellen und 2 Diagramme erstellen.

Beabsichtigte Lernergebnisse

- Das beabsichtigte Ziel dieses Berichts ist es, dass Sie nachweisen:

Kenntnisse und Fähigkeiten

- die Fähigkeit, zusammenfassende Statistiken zu berechnen
- die Fähigkeit, Tabellen mit zusammenfassenden Statistiken und Datenverteilungen zu erstellen

Kommunikation

- die Fähigkeit, formatierte Tabellen mit zusammenfassenden Statistiken zu drucken
- die Fähigkeit, Unterschiede zwischen Datengruppen durch Bezugnahme auf zusammenfassende Statistiken und Diagramme zu beschreiben

Bewertung

Die Bewertung erfolgt auf der Grundlage der Replikation der

- die Struktur des Dokuments (die HTML-Ausgabe)
- die Struktur und Genauigkeit des Codes (.qmd-Quellcode)

- die Ähnlichkeit zwischen den Originalplots und Ihren Plots

Tipps

Einige Tipps:

- Sie können auf Deutsch oder Englisch schreiben, je nachdem, was Sie bevorzugen
- Ich empfehle Ihnen dringend, Ihr Dokument *oft* zu rendern! Auf diese Weise können Sie, wenn ein Problem auftaucht, besser einschätzen, welcher Code es verursacht.
- Wenn Sie das Gefühl haben, dass Sie etwas langsam sind, versuchen Sie, sich auf das “große Ganze” zu konzentrieren und machen Sie sich keine Gedanken über all die Details in den Plots
 - Versuchen Sie zunächst, den Plot im Allgemeinen zu erstellen, und gehen Sie dann zu den Details über
 - Ich persönlich würde zuerst versuchen, die Handlung im Großen und Ganzen zu entwerfen, und mich dann um die Details kümmern

Tun Sie einfach Ihr Bestes!

Biondo, Soilemezidi, und Mancini (2022): Verarbeitung der Kongruenz der Zeitformen der Verben

Wir werden uns wieder mit den Daten von Biondo, Soilemezidi, und Mancini (2022) arbeiten. Sie müssen die Daten nicht vollständig verstehen, um die Aufgaben zu lösen, aber vielleicht finden Sie sie hilfreich. Ihre Hauptaufgabe besteht darin, vier Diagramme zur Untersuchung der Daten zu erstellen.

Unsere Daten stammen aus einem Eye-Tracking-Experiment von Biondo, Soilemezidi, und Mancini (2022), bei dem:

- Die Teilnehmer lasen (spanische) Sätze mit Zeitformeln aus der Vergangenheit oder Zukunft (z. B. *ayer* ‘gestern’ oder *mañana* ‘morgen’) und Verbformen aus der Vergangenheit oder Zukunft (z. B. *compraron* ‘gekauft’ oder *comprarán* ‘wird kaufen’). In Abbildung 1 finden Sie Beispielsätze in der Vergangenheit und in der Zukunft unter grammatikalischen (“match”) und ungrammatikalischen (“mismatch”) Bedingungen.
- Die Augenbewegungen wurden aufgezeichnet, während die Teilnehmer die Sätze lasen
- Die Haupthypothese: Wenn die Temporalphrasen nicht mit der Verbform übereinstimmen (z.B. ‘gestern...kaufen’ oder ‘morgen...kaufen’), sollten *längere* Lesemaßnahmen in der Verbregion beobachtet werden.
- Eine weitere Frage: Wird dieser Effekt sowohl für die Vergangenheit als auch für die Zukunft beobachtet?

Table 3
Experimental Material Addressing Question 1

Past	
Match	Mismatch
Gracias a la beca, el año pasado los investigadores <u>progresaron</u> en sus estudios sobre la polución. “Thanks to the scholarship, last year the researchers made progress on their studies on pollution”	Gracias a la beca, el año pasado los investigadores <u>progresarán</u> en sus estudios sobre la polución. “Thanks to the scholarship, last year the researchers will make progress on their studies on pollution”
Future	
Match	Mismatch
Gracias a la beca, el próximo año los investigadores <u>progresarán</u> en sus estudios sobre la polución. “Thanks to the scholarship, next year the researchers will make progress on their studies on pollution”	Gracias a la beca, el próximo año los investigadores <u>progresaron</u> en sus estudios sobre la polución. “Thanks to the scholarship, next year the researchers made progress on their studies on pollution”

Note. The underlined terms represent the target region.

Abbildung 1: Beispielsätze aus Biondo, Soilemezidi, und Mancini (2022)

Variablen

Unsere **Messvariablen** (d. h. abhängige Variablen) sind drei eye-tracking Messungen der Lesezeit:

- Lesezeit beim ersten Durchgang (*first-pass reading time*: Summe aller Fixierungen auf einer Region von Interesse, bevor diese entweder nach links oder nach rechts verlassen wird)
- Regressionspfaddauer (*regression path duration*: Summe des ersten Durchlaufs plus der Zeit, die für das erneute Lesen früherer Regionen aufgewendet wird, bevor die Region von Interesse passiert wird)
- Gesamtlesedauer (*total reading time*: die Gesamtzeit, die beim Lesen des Satzes auf das Verb geschaut wird)

Unsere **Prädiktorvariablen** (d. h. unabhängige Variablen) sind:

- die *Temporaladverb Zeitbezug* (Vergangenheit oder Zukunft)
- die *Verbform (tense)* (Vergangenheit oder Zukunft)
- die *Satzregion* (Adverb oder Verb)

Vorhersagen

In der Regel gehen wir davon aus, dass längere Lesezeiten auf *Verarbeitungsschwierigkeiten*, d. h. Schwierigkeiten bei der Sprachverarbeitung, hinweisen. Die Hypothese war daher, dass längere Lesezeiten für die *Verb*-Region in ungrammatischen Sätzen vorhanden sein würden. Diese Frage werden wir heute mit den Daten nicht beantworten können.

1 Einrichten

1.1 YAML

- Erstellen Sie ein neues Quarto-Dokument
- Stellen Sie sicher, dass Ihre YAML enthält:
 - einen passenden Titel
 - Ihren Vor- und Nachnamen (`author`)
 - das Datum
 - das Inhaltsverzeichnis
 - unter `format: html...`, `self-contained: true` einfügen
 - * und setzen Sie die globalen Chunk-Optionen auf `echo: false`
- Ihr YAML sollte etwa so aussehen

```
---
title: "Summary assignment: wrangling and data viz"
subtitle: "In-class assignment 1"
author: "YOUR NAME HERE"
institute: Humboldt-Universität zu Berlin
lang: de
date: "`r Sys.Date()`"
format:
  html:
    toc: true
    number-sections: true
    self-contained: true
    execute:
      echo: false
---
```

1.2 Struktur des Dokuments

- Sie werden Ihr Quarto-Skript und die HTML-Ausgabe Ihres Skripts einreichen.
- Um die Lesbarkeit des gerenderten Dokuments zu verbessern, sollten Sie das Dokument durch Überschriften und Zwischenüberschriften strukturieren.
- Verwenden Sie schriftlichen Text, wo es notwendig ist, um Fragen zu beantworten/ Prozesse zu beschreiben
- Verwenden Sie bei Bedarf Code-Bausteine
- eine gute Strategie für die Wahl einer Überschrift: Wenn dieses Dokument einen neuen Abschnitt enthält, sollten Sie auch einen neuen Abschnitt erstellen

1.3 Pakete

- werden Sie die Pakete benötigen:
 - tidyverse
 - here
 - janitor
 - devtools
 - knitr
 - kableExtra

Laden Sie sie mit dem Paket `pacman` (Funktion `p_load()`).

```
pacman::p_load(tidyverse,
               here,
               janitor,
               devtools,
               knitr,
               kableExtra)
```

```
pacman::p_load(tidyverse,
               here,
               ggthemes,
               patchwork,
               devtools,
               knitr,
               kableExtra)
```

1.4 Datenbeschreibung

1. Laden Sie den Datensatz `Biondo_etal_2021_bericht2.csv` aus Moodle herunter, und speichern Sie ihn im Ordner `daten`.
2. Dann laden Sie die Daten und verwenden die Funktion `clean_names()` aus dem Paket `janitor`, um die Variablennamen zu bereinigen.

Unsere Variablen sind in Tabelle 1 beschrieben.

```
df_biondo <- read_csv(here("daten", "Biondo_etal_2021_bericht2.csv")) %>%
  janitor::clean_names()
```

Tabelle 1: Variable names and descriptions for dataset Biondo_etal_2021_bericht2.csv

variable	type	description
subj	numeric	Participant ID
item_num	numeric	Item number
label	character	sentence region text
total_time	numeric	total reading time in milliseconds (eye-tracking: sum of all fixations made on the sentence)
adverb_time	character	adverb time reference: past/future
gramm	numeric	grammaticality: whether the adverb time and verb tense matched (0: ungrammatical, 1: grammatical)
verb_tense	character	verb tense: past/future
first_pass	numeric	first-pass reading time in milliseconds (eye-tracking: sum of all fixations on the sentence during the first pass)
regression_path	numeric	regression path duration/go-past time (eye-tracking: sum of the first-pass path duration and the regression path duration)
regress_in	numeric	regression in; 0 = no, 1 = yes (eye-tracking: whether the region received a regression in)
regress_out	numeric	regression out; 0 = no, 1 = yes (eye-tracking: whether a regression out of the sentence region occurred)
region	character	sentence region; here we only have adverb/verb

2 Data wrangling

1. Ordnen (d.h., `arrange()`) Sie die Daten nach Thema, Positionsnummer und Region, und dann,
2. Herausfiltern (d.h. `filter()`) von Beobachtungen, bei denen `adverb_time` `none` ist, and then,
3. Ordne die Variablen neu an (d.h. `relocate()`), so dass sie in der folgenden Reihenfolge stehen:

- subj, item_num, region, label, adverb_time, verb_tense, gramm, first_pass, regression_path, total_time, regress_in, regress_out

```
df_biondo <- df_biondo %>%
  filter(adverb_time != "none") %>%
  arrange(subj, item_num, region) %>%
  relocate(subj, item_num, region, label, adverb_time, verb_tense, gramm, first_pass, r
```

2. Drucken Sie die ersten 10 Zeilen Ihres Datenrahmens mit den Funktionen `kable()` und `kable_styling()`. Dies sollte wie Tabelle 2 aussehen.

```
df_biondo %>%
  head(n = 10) %>%
  kable() %>%
  kable_styling()
```

Tabelle 2: First 10 rows of df_biondo after data wrangling

subj	item_num	region	label	adverb_time	verb_tense	gramm	first_pass	reg
1	1	adverb	mañana	Future	Future	1	453	
1	1	verb	representarán	Future	Future	1	1596	
1	2	adverb	el próximo domingo	Future	Future	1	782	
1	2	verb	alzarán	Future	Future	1	316	
1	3	adverb	el mes que viene	Future	Future	1	614	
1	3	verb	centrarán	Future	Future	1	238	
1	4	adverb	el próximo año	Future	Future	1	593	
1	4	verb	coleccionarán	Future	Future	1	562	
1	5	adverb	mañana a la una	Future	Future	1	726	
1	5	verb	complementarán	Future	Future	1	195	

2.1 Data tidying

1. Erstellen Sie eine längere Version des Datensatzes, so dass die Namen der Spalten `first_pass`, `regression_path` und `total_time` in einer neuen Variablen `measure` und ihre Werte in einer neuen Variablen `time` gespeichert werden. Sie müssen die Funktion `pivot_longer()` verwenden.

```
df_biondo_long <-
  df_biondo %>%
  pivot_longer(
    cols = c(first_pass, regression_path, total_time),
    names_to = "measure",
    values_to = "time"
  ) %>%
  relocate(measure, time, .after=gramm)
```

2. Drucken Sie die ersten zehn Zeilen des neuen Datenrahmens mit den Funktionen `kable()` und `kable_styling()`. Es sollte wie Tabelle 3 aussehen.

```
df_biondo_long %>%
  head(n = 10) %>%
  kable() %>%
  kable_styling()
```

3 Plot-Interpretation

In diesem Abschnitt werden Sie verschiedene Arten von Diagrammen interpretieren. Tipp: Lesen Sie immer zuerst den Titel und/oder die Beschriftung einer Tabelle und sehen Sie

Tabelle 3: First 10 rows of df_biondo_longer

subj	item_num	region	label	adverb_time	verb_tense	gramm	measure
1	1	adverb	mañana	Future	Future	1	first_pass
1	1	adverb	mañana	Future	Future	1	regression_path
1	1	adverb	mañana	Future	Future	1	total_time
1	1	verb	representarán	Future	Future	1	first_pass
1	1	verb	representarán	Future	Future	1	regression_path
1	1	verb	representarán	Future	Future	1	total_time
1	2	adverb	el próximo domingo	Future	Future	1	first_pass
1	2	adverb	el próximo domingo	Future	Future	1	regression_path
1	2	adverb	el próximo domingo	Future	Future	1	total_time
1	2	verb	alzarán	Future	Future	1	first_pass

sich an, was die y- und x-Achse darstellen, sowie alle anderen ästhetischen Merkmale (z. B. die Farbe).

3.1 Distributions

1. Vergleichen Sie den Code für fig_1, fig_2 und fig_3 mit der Ausgabe von Abbildung 2. Welcher Code ergibt A, B, und C? Wie können Sie das erkennen?

```
fig_1 <-
  df_biondo_long %>%
  ggplot(aes(x = time, fill = verb_tense, colour = verb_tense)) +
  labs(colour = "Tense",
        fill = "Tense") +
  facet_wrap(~region) +
  geom_histogram(alpha = .2) +
  theme_bw()
```

```
fig_2 <-
  df_biondo_long %>%
  ggplot(aes(x = time, fill = verb_tense, colour = verb_tense)) +
  labs(x = "Time (ms)",
        y = "Number of observations") +
  facet_wrap(~measure) +
  geom_histogram(alpha = .2) +
  theme_bw()
```

```
fig_3 <-
  df_biondo_long %>%
  ggplot(aes(x = time, y = adverb_time, fill = verb_tense, colour = verb_tense)) +
```

```
labs(x = "Time (ms)",  
     y = "Adverb time reference") +  
facet_wrap(~measure) +  
geom_boxplot(alpha = .2) +  
theme_bw()  
  
library(patchwork)  
(fig_3 /  
  fig_1 /  
    fig_2) +  
plot_annotation(tag_levels = "A")
```

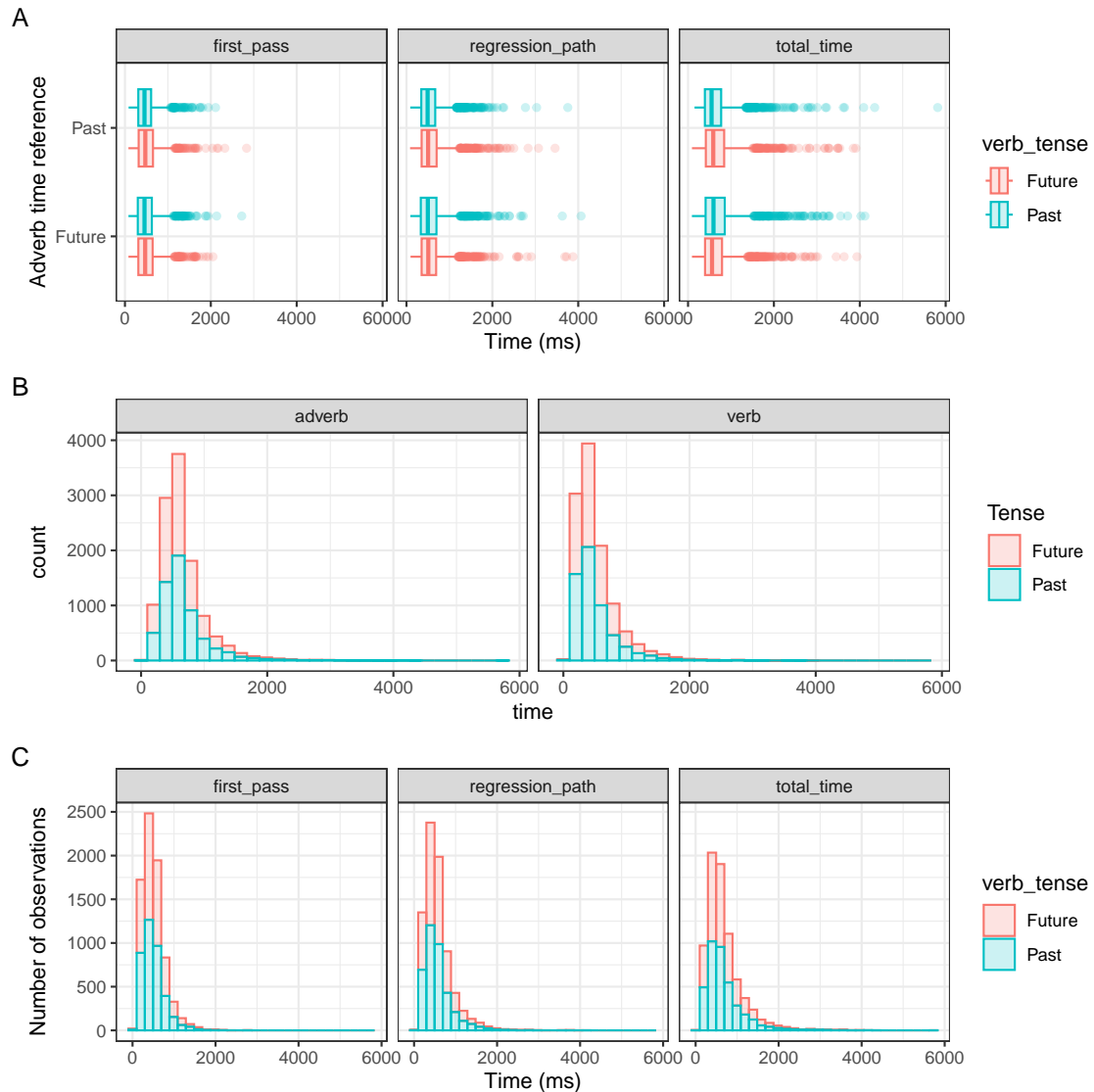


Abbildung 2: Histogram 1 and Histogram 2

2. Welche Darstellung(en) in Abbildung 2 zeigen die *Modus*-Lesezeiten?
3. Welche Darstellung(en) in Abbildung 2 zeigen die *mittleren* Lesezeiten?
4. Welche Darstellung(en) in ?@fig-Histogrammen zeigt/zeigen den *Bereich* der Ablesezeiten?

3.2 Zusammenfassende Statistiken

1. Schau dir Abbildung 3 an.
 - Welche drei **geom** werden benötigt, um diese Darstellung zu erstellen?
 - Was stellt das linke obere Feld dar (z. B. Maßnahme, Satzregion)?

- Welche Verbform hatte eine längere Regressionspfaddauer in der Verbregion?
- Beschreiben Sie das Muster der Gesamtlesezeit in der Verbregion (d. h., welche Unterschiede stellen Sie zwischen den Bedingungen fest?).

```

labs_measure = c(
  "first_pass" = "First-pass",
  "regression_path" = "Regression path",
  "total_time" = "Total time",
  "adverb" = "Adverb",
  "verb" = "Verb"
)

pd = position_dodge(0.2)
df_biondo_long %>%
  drop_na() %>%
  summarise(N = n(),
            mean = mean(time),
            sd = sd(time),
            upper = mean+sd,
            lower = mean-sd,
            .by = c(region, adverb_time, verb_tense, gramm, measure)) %>%
  arrange(region, adverb_time, desc(gramm), measure) %>%
  ggplot(aes(x = as_factor(gramm), y = mean, colour = verb_tense)) +
  facet_grid(region~measure, labeller = as_labeller(labs_measure)) +
  labs(y = "Reading time (ms)",
       x = "Grammaticality",
       colour = "Verb tense",
       linetype = "Verb tense") +
  geom_point(position = pd) +
  geom_errorbar(aes(ymin=lower, ymax=upper), width = 0.2, position = pd) +
  geom_line(aes(group = verb_tense, linetype = verb_tense), position = pd) +
  theme_bw()

```

Tabelle 4: ?(caption)

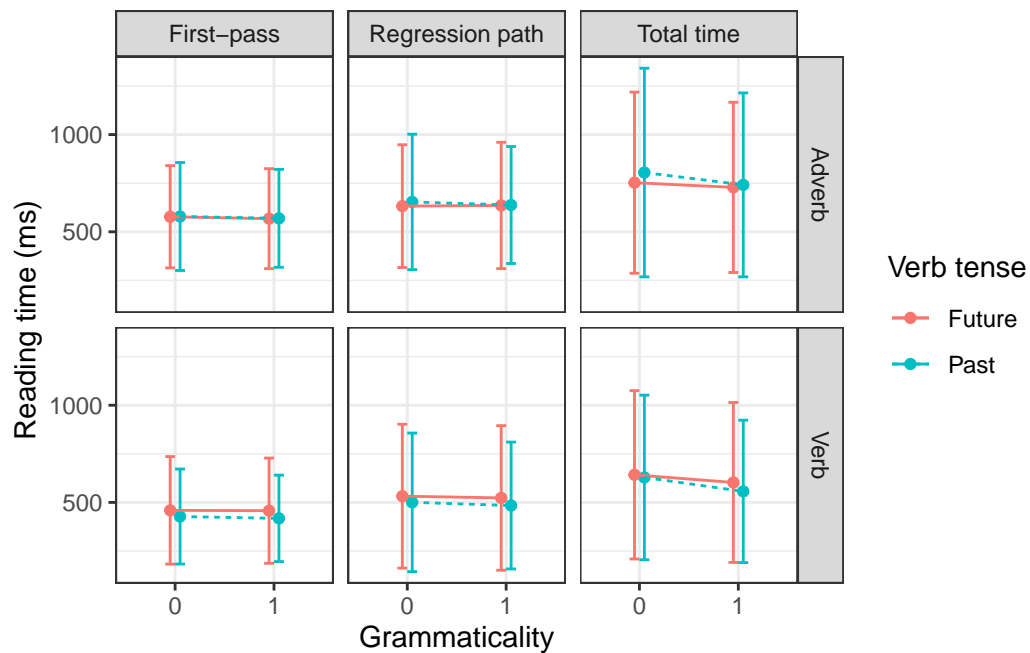


Abbildung 3: Mean reading times per condition (with +/- 1SD)

4 Data summaries

1. Erstellen Sie eine Tabelle wie `?@tbl-summarise` aus dem Datenrahmen `df_biondo_long`. Sie müssen die Funktion `summarise()` verwenden, um die Anzahl der Beobachtungen (`n()`), den Mittelwert (`mean()`) und die Standardabweichung (`sd()`) zu berechnen. Denken Sie daran, die Daten nach (`.by =`) zu gruppieren, welche Variablen auch immer in `?@tbl-summarise` angegeben sind.

```
sum_region_et <-
  df_biondo_long %>%
  drop_na() %>%
  summarise(N = n(),
            mean = mean(time),
            sd = sd(time),
            .by = c(region, adverb_time, verb_tense, measure)) %>%
  arrange(region, adverb_time, measure)
```

Tabelle 5: Mean and standard deviations per region and condition

region	adverb_time	verb_tense	measure	N	mean	sd
adverb	Future	Future	first_pass	953	567.6107	257.1014
adverb	Future	Past	first_pass	953	578.6967	277.6434
adverb	Future	Future	regression_path	953	635.6002	324.8907
adverb	Future	Past	regression_path	953	653.6128	348.4972
adverb	Future	Future	total_time	953	728.5278	437.8639
adverb	Future	Past	total_time	953	804.8374	536.7249
adverb	Past	Past	first_pass	954	569.2610	251.8370
adverb	Past	Future	first_pass	953	577.3820	263.0772
adverb	Past	Past	regression_path	954	637.8847	300.9607
adverb	Past	Future	regression_path	953	632.0588	315.7681
adverb	Past	Past	total_time	954	741.4140	473.1569
adverb	Past	Future	total_time	953	752.6306	465.9965
verb	Future	Future	first_pass	948	457.4409	270.6509
verb	Future	Past	first_pass	945	427.3079	244.4802
verb	Future	Future	regression_path	948	523.1719	371.8651
verb	Future	Past	regression_path	945	500.5524	356.4027
verb	Future	Future	total_time	948	602.7521	411.5918
verb	Future	Past	total_time	945	628.4836	423.4241
verb	Past	Past	first_pass	953	417.9307	222.3428
verb	Past	Future	first_pass	949	459.2160	276.7817
verb	Past	Past	regression_path	953	484.0598	326.3804
verb	Past	Future	regression_path	949	532.1349	369.9950
verb	Past	Past	total_time	953	556.7817	366.4723
verb	Past	Future	total_time	949	642.0242	433.0259

```
sum_region_et %>%
  knitr::kable() %>%
  kableExtra::kable_styling()
```

5 Plotten der Daten

In diesem Abschnitt werden Sie 2 Diagramme erstellen.

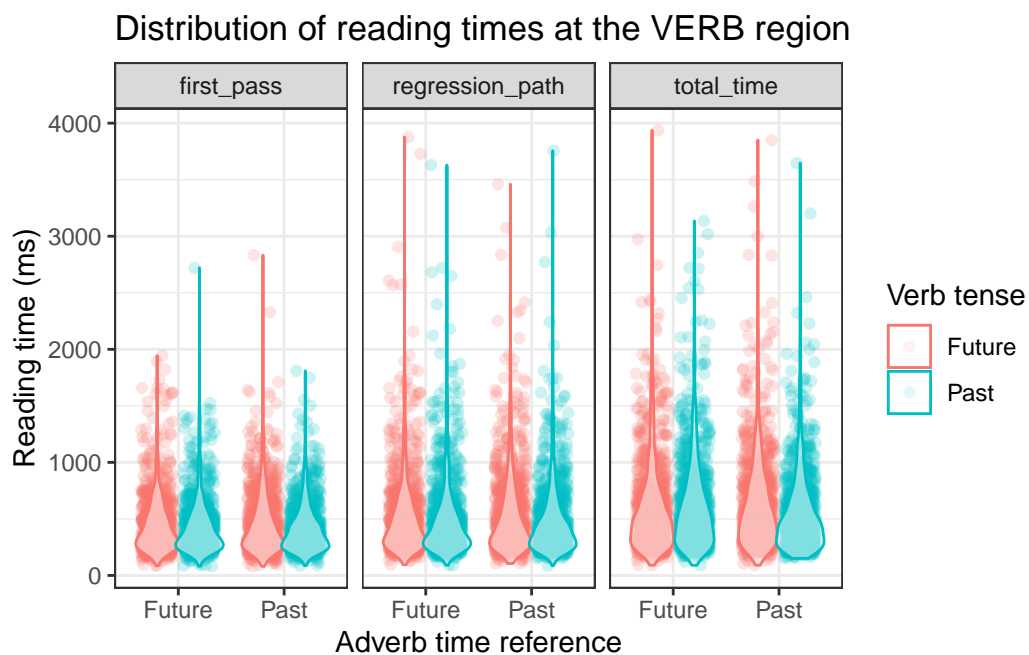
5.1 Verteilungen

1. Erzeugen Sie Abbildung 4. Du musst verwenden:

- `df_biondo_long`
- `filter()` (nur Verbregion)

- 2 geoms
- facet_wrap(~)
- labs()
- position_jitterdodge()
- alpha = (zur Kontrolle der Transparenz eines Geoms)

```
pjd <- position_jitterdodge(jitter.width=.3, dodge.width= .8)
df_biondo_long %>%
  filter(region == "verb") %>%
  ggplot(aes(x = adverb_time, y = time, colour = verb_tense)) +
  facet_wrap(~measure) +
  labs(title = "Distribution of reading times at the VERB region",
       x = "Adverb time reference",
       y = "Reading time (ms)",
       colour = "Verb tense") +
  geom_point(alpha = .2, position = pjd) +
  geom_violin(alpha = .5, position = position_dodge(.8)) +
  theme_bw()
```



5.2 Split-violin plots

Für diese Aufgabe werden Sie eine neue Art von Diagramm erstellen: das Split-Violine-Diagramm. Dies ist derzeit nicht einfach mit `ggplot2` zu machen, also müssen wir ein neues

Paket installieren: `introdaviz`. Dieses Paket begleitet Nordmann u. a. (2022), und hat auch ein [Online-Tutorial](#).

1. Installieren Sie `introdaviz` (dies funktioniert nur, wenn Sie bereits das Paket `devtools` mit `pacman::p_load()` am Anfang dieses Skripts installiert/geladen haben)

```
# how to install the introdaviz package to get split and half violin plots
devtools::install_github("psyteachr/introdaviz")
```

2. Einladen von `Introdaviz`

```
library(introdaviz)
```

3. Erstellen Sie eine mehrschichtige Darstellung mit:

- Datenrahmen `df_biondo_long`
- Zeit" auf der y-Achse
- `verb_tense` auf der x-Achse
- `gramm` colour und fill (Achtung: `gramm` muss `as_factor()` gelesen werden)
- Messung" in `facet_wrap(~)`
- split-violin plot (`geom_split_violin()` aus `introdaviz`)
- Boxplot (`Geom_boxplot()` aus `ggplot2`)

Tipp: Die Erstellung dieses Diagramms erfolgt auf genau dieselbe Weise, als ob Sie einfach `geom_violin()` anstelle von `geom_split_violin()` verwenden würden, um ein Violin-Boxplot-Diagramm zu erstellen.

3. Korrigieren Sie dann die Beschriftungen und passen Sie die Transparenz des `geoms` an (mit `alpha()`), so dass es ähnlich aussieht wie Abbildung 5.

```
df_biondo_long %>%
  ggplot(aes(x = verb_tense, y = log(time), colour = as_factor(gramm), shape = as_factor(gramm))) +
  facet_wrap(~measure) +
  labs(title = "Distribution of eye-tracking during reading measures per condition",
       x = "Verb Tense",
       y = "Reading time (ms)",
       colour = "Grammaticality",
       shape = "Grammaticality",
       fill = "Grammaticality") +
  geom_split_violin(alpha = .2) +
  geom_boxplot(alpha = .2, aes(shape = as_factor(gramm))) +
  theme_bw()
```

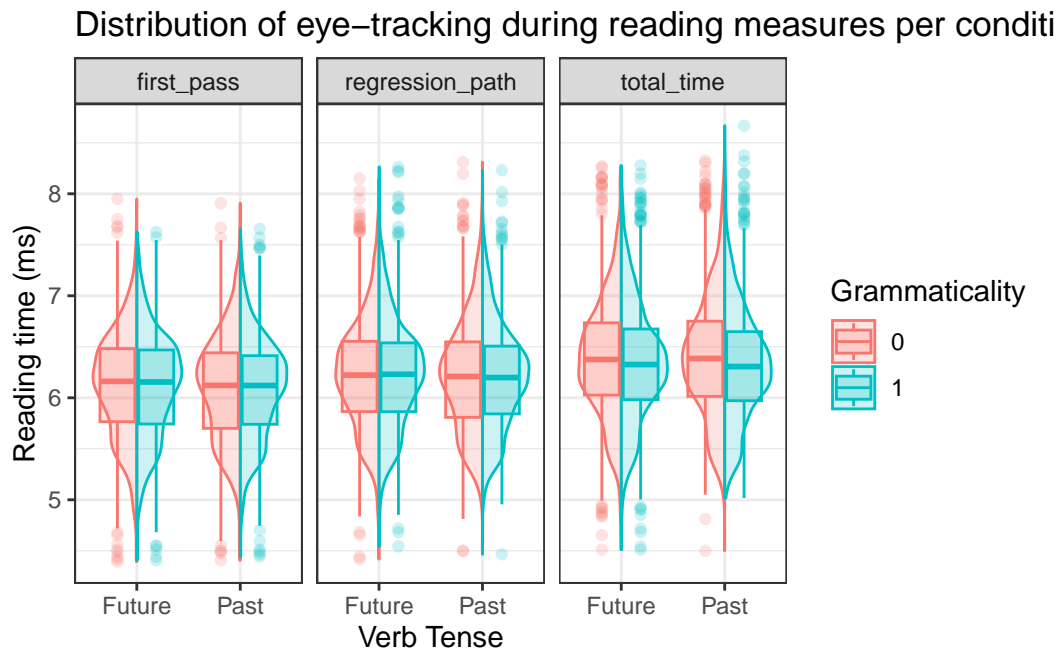



Abbildung 5: A split-violin boxplot

6 Code-Chunk-Optionen

Für diesen Bericht haben Sie zwei Tabellen gedruckt und zwei Diagramme erstellt. Wenn Sie dies noch nicht getan haben, erstellen Sie entsprechende Beschriftungen (`#| label: tbl-` oder `fig-`) und Überschriften (`tbl-cap:` oder `fig-cap:`) für diese Tabellen und Abbildungen.

7 Session Info einbeziehen

Am Ende Ihres Berichts:

1. Erstellen Sie einen neuen Abschnitt (`# Session Info`)
2. Fügen Sie den folgenden Text als Inline-Text ein:

Hergestellt mit ``r R.version.string`` (``r R.version$nickname``).

3. Fügen Sie das Folgende in einen Codeabschnitt ein:

```
sessionInfo()
```

Wenn Sie nun Ihr Dokument rendern, sollte eine Zusammenfassung der geladenen Pakete und ihrer Versionen angezeigt werden.

8 Bericht einreichen

Wenn Sie fertig sind, laden Sie Ihr `.qmd`-Skript *und* die gerenderte Ausgabe im HTML-Format in Moodle hoch.

Session Info

Hergestellt mit R version 4.3.0 (2023-04-21) (Already Tomorrow) und RStudioversion 2023.3.0.386 (Cherry Blossom).

```
sessionInfo()
```

R version 4.3.0 (2023-04-21)

Platform: aarch64-apple-darwin20 (64-bit)

Running under: macOS Ventura 13.2.1

Matrix products: default

BLAS: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib

LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;

locale:

[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

time zone: Europe/Berlin

tzcode source: internal

attached base packages:

[1] stats graphics grDevices utils datasets methods base

other attached packages:

[1] introdataviz_0.0.0.9003	patchwork_1.1.2	ggthemes_4.2.4
[4] kableExtra_1.3.4.9000	knitr_1.43	devtools_2.4.5
[7] usethis_2.1.6	janitor_2.2.0	here_1.0.1
[10] lubridate_1.9.2	forcats_1.0.0	stringr_1.5.0
[13] dplyr_1.1.2	purrr_1.0.1	readr_2.1.4
[16] tidyr_1.3.0	tibble_3.2.1	ggplot2_3.4.2
[19] tidyverse_2.0.0		

loaded via a namespace (and not attached):

[1] tidyselect_1.2.0	viridisLite_0.4.2	farver_2.1.1	fastmap_1.1.1
[5] pacman_0.5.1	promises_1.2.0.1	digest_0.6.31	timechange_0.2.0
[9] mime_0.12	lifecycle_1.0.3	ellipsis_0.3.2	processx_3.8.1
[13] magrittr_2.0.3	compiler_4.3.0	rlang_1.1.1	tools_4.3.0

[17]	utf8_1.2.3	yaml_2.3.7	prettyunits_1.1.1	labeling_0.4.2
[21]	htmlwidgets_1.6.2	curl_5.0.1	bit_4.0.5	pkgbuild_1.4.0
[25]	plyr_1.8.8	xml2_1.3.4	pkgload_1.3.2	miniUI_0.1.1.1
[29]	withr_2.5.0	grid_4.3.0	fansi_1.0.4	urlchecker_1.0.1
[33]	profvis_0.3.8	xtable_1.8-4	colorspace_2.1-0	scales_1.2.1
[37]	cli_3.6.1	rmarkdown_2.22	crayon_1.5.2	generics_0.1.3
[41]	remotes_2.4.2	rstudioapi_0.14	httr_1.4.6	tzdb_0.4.0
[45]	sessioninfo_1.2.2	cachem_1.0.8	rvest_1.0.3	parallel_4.3.0
[49]	vctrs_0.6.3	webshot_0.5.4	jsonlite_1.8.5	callr_3.7.3
[53]	hms_1.1.3	bit64_4.0.5	systemfonts_1.0.4	glue_1.6.2
[57]	ps_1.7.5	stringi_1.7.12	gtable_0.3.3	later_1.3.1
[61]	munsell_0.5.0	pillar_1.9.0	htmltools_0.5.5	R6_2.5.1
[65]	rprojroot_2.0.3	vroom_1.6.3	evaluate_0.21	shiny_1.7.4
[69]	memoise_2.0.1	snakecase_0.11.0	httpuv_1.6.11	Rcpp_1.0.10
[73]	svglite_2.1.1	xfun_0.39	fs_1.6.2	pkgconfig_2.0.3

Literaturverzeichnis

- Biondo, Nicoletta, Marielena Soilemezidi, und Simona Mancini. 2022. „Yesterday Is History, Tomorrow Is a Mystery: An Eye-Tracking Investigation of the Processing of Past and Future Time Reference During Sentence Reading.“ *Journal of Experimental Psychology: Learning, Memory, and Cognition* 48 (7): 1001–18. <https://doi.org/10.1037/xlm0001053>.
- Nordmann, Emily, Phil McAleer, Wilhelmiina Toivo, Helena Paterson, und Lisa M. DeBruine. 2022. „Data Visualization Using R for Researchers Who Do Not Use R.“ *Advances in Methods and Practices in Psychological Science* 5 (2): 251524592210746. <https://doi.org/10.1177/25152459221074654>.