Datenvisualiserung 1

Visualisierung von Verteilungen

Daniela Palleschi
Humboldt-Universität zu Berlin

Mi. den 25.10.2023

Überblick

- Heutige Ziele
- Datenrahmen
- Lexical Decision Task (LDT)
- `lexdec` Datensatz
- Erstellen von Plots mit `ggplot2`
- Entscheidung für ein Geom
- Exercises
- Session Info

Wiederholung

Letzte Woche haben wir…

- R und RStudio installiert
- unser erstes R-Skript erstellt
- einfache Arithmetik mit Objekten und Vektoren durchgeführt

Wiederholung

```
1 x <- c(1,2,3)
2 y <- sum(1,2,3)
```

- Was enthalten die Vektoren `x` und `y`?
- Das Objekt `x` enthält `1, 2, 3`
- Das Objekt `y` enthält `` 6 ``

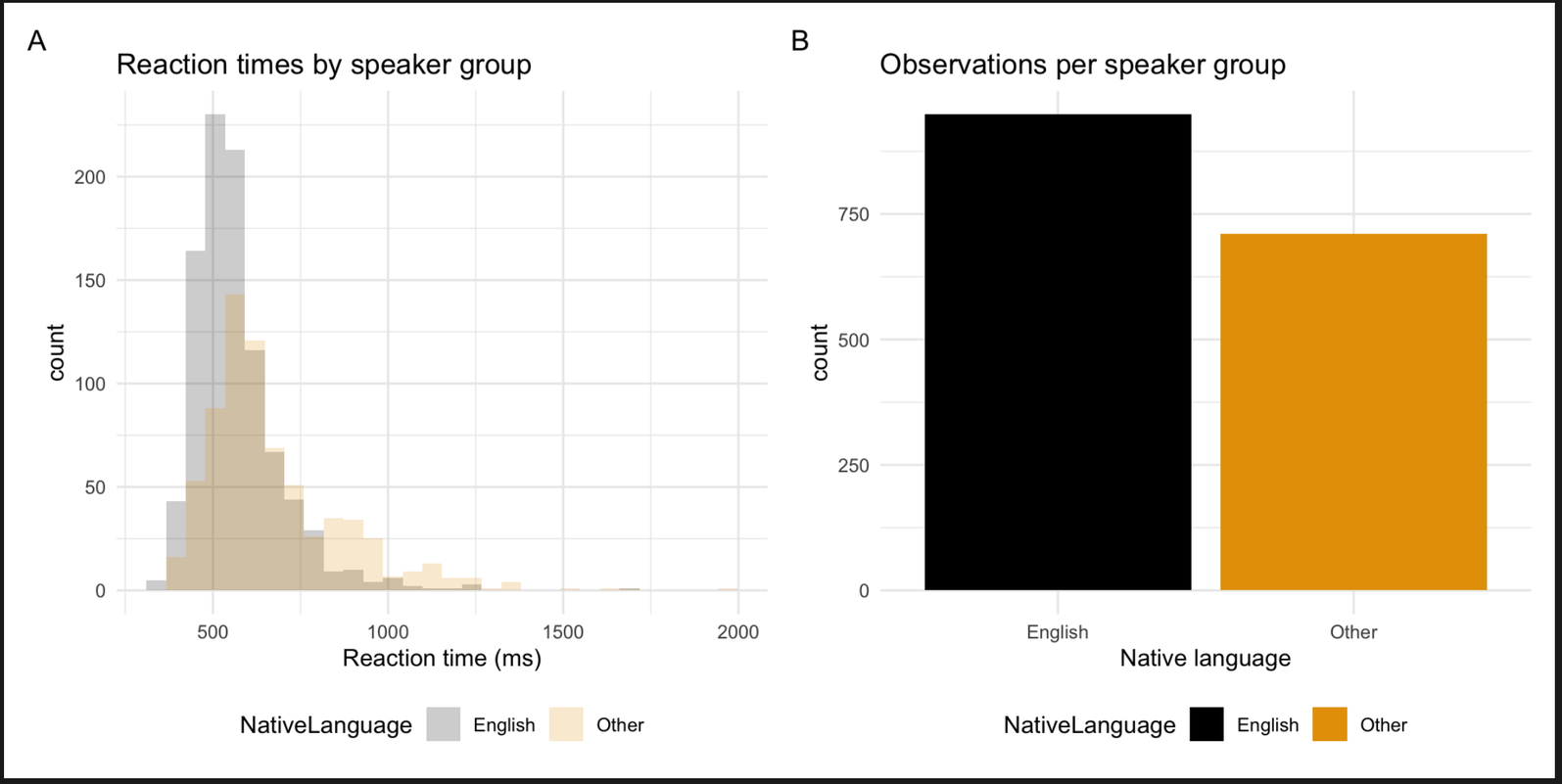
Heutige Ziele

Heute werden wir lernen…

- was Datenframes sind
- den Unterschied zwischen kategorialen und kontinuierlichen Daten
- wie man Diagramme mit `ggplot` erstellt
- die richtige Darstellung für unsere Daten auszuwählen

Endgültiges Ziel

- Unser heutiges Ziel ist es, die Daten wie folgt zu visualisieren
  - Das Diagramm zeigt die Verteilung (`Anzahl`) der Reaktionszeiten und der Muttersprache der Teilnehmer



Lust auf mehr?

- Kapitel 2 (Datenvisualisierung) in Wickham et al. (2023), bis zum Abschnitt 2.4
- Kapitel 3 (Datenvisualisierung) in Nordmann & DeBruine (2022)

Vorbereitung

In Ihrem RProject-Ordner…

- erstellen Sie einen neuen Ordner mit dem Namen `moodle`
  - Laden Sie die Moodle-Materialien von heute herunter und speichern Sie sie dort
- Erstellen Sie einen neuen Ordner in `notes` mit dem Namen `02-datenviz1`
- öffne ein neues `.R` Skript
  - speichere es in dem neuen Ordner

Pakete

- Pakete laden (und installieren)
  - `tidyverse`
  - `languageR`
  - `ggthemes`
  - `patchwork`

```
# in the CONSOLE: install packages if needed
install.packages("tidyverse")
install.packages("languageR")
install.packages("ggthemes") # for customising our plots
install.packages("patchwork") # plot layouts
```

```
# Pakete laden
library(tidyverse)
library(languageR)
library(ggthemes)
library(patchwork)
```

Datenrahmen

- Datenrahmen sind eine Sammlung von Variablen, wobei
  - jede Variable eine Spalte ist
  - jede Zeile eine einzelne Beobachtung/ein einzelner Datenpunkt ist
  - jede Zelle in einer Zeile verknüpft ist
- Datenrahmen sind genau wie Tabellenkalkulationen, aber rechteckig
- Verschiedene Wörter für Datenrahmen:
  - Datenrahmen
  - Datensatz
  - Tibble (im `tidyverse`)

Sprechen über Datensätze

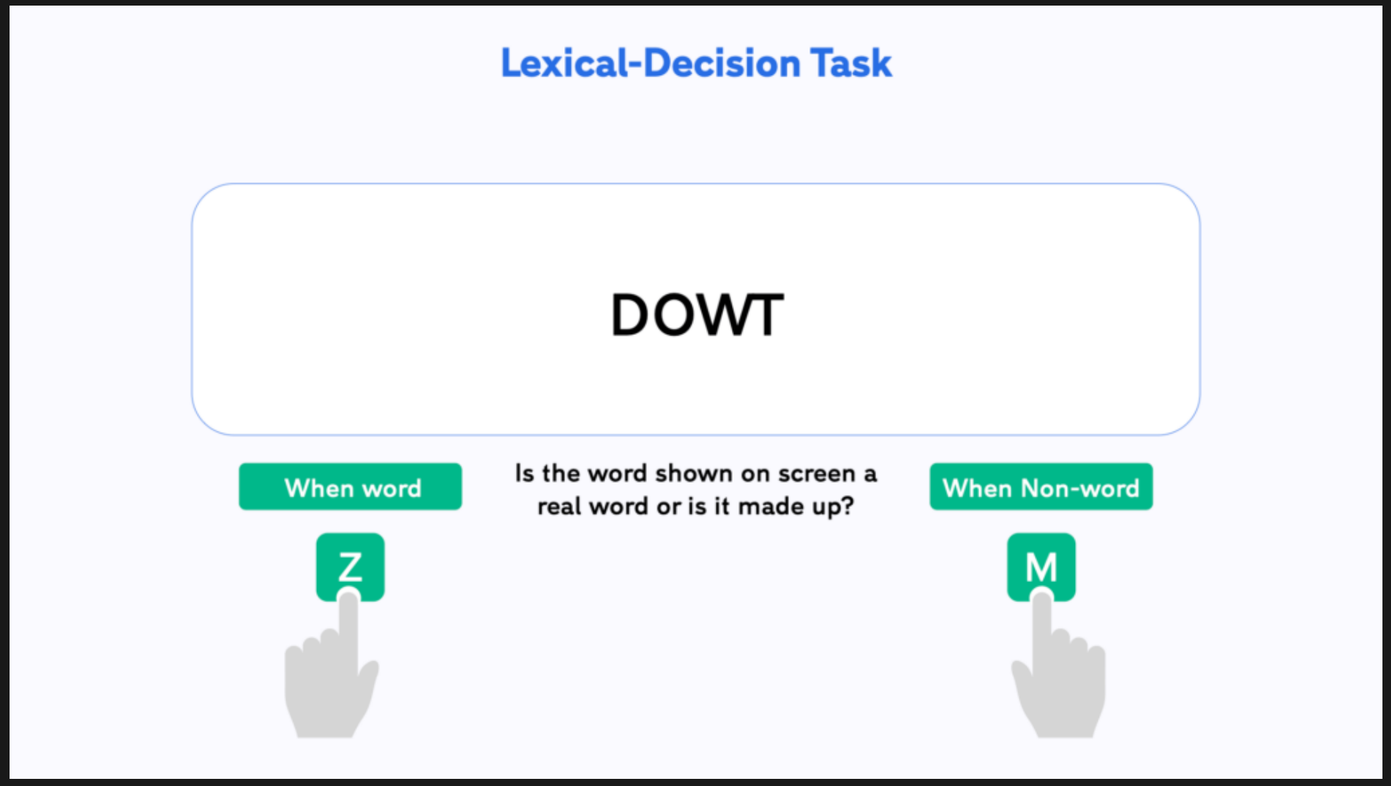
- eine **Variable**: eine Menge, Qualität oder Eigenschaft, die man messen kann
- ein **Wert**: der Zustand einer Variablen, wenn man sie misst
- eine **Beobachtung**: eine Reihe von Messungen, die unter ähnlichen Bedingungen durchgeführt werden
  - enthält mehrere Werte, die jeweils mit einer Variablen verbunden sind
  - eine Beobachtung für eine einzelne Variable wird manchmal als *Datenpunkt* bezeichnet
- **Tabellendaten** sind eine Reihe von Werten, die jeweils mit einer Variablen und einer Beobachtung verbunden sind
  - Tabellarische Daten sind “tidy”, wenn jeder Wert in einer eigenen *Zelle*, jede Variable in einer eigenen Spalte und jede Beobachtung in einer eigenen Zeile steht

Kategoriale und kontinuierliche Variablen

- Wie wir die Verteilung einer Variablen darstellen, hängt davon ab, welche Art von Daten sie repräsentiert: *kategorisch* oder *numerisch*
- Eine Variable ist *kategorisch*, wenn sie eine kleine Menge von Werten annehmen kann, die sich in Gruppen zusammenfassen lassen
  - z. B. alt/jung, klein/groß, grammatikalisch/ungrammatikalisch, L1/L2-Sprecher
- eine Variable ist *numerisch* (d. h. quantitativ), wenn sie eine große Bandbreite an numerischen Werten annehmen kann
  - und es sinnvoll wäre, zu addieren, zu subtrahieren, den Mittelwert zu berechnen usw.
  - kann *kontinuierlich* sein (Dezimalpunkte sind sinnvoll, z. B. 1,5 cm)
  - oder *diskret* (Dezimalpunkte sind *nicht* sinnvoll, z. B. 1,5 Kinder sind nicht sinnvoll)
- wir erstellen verschiedene Diagramme, je nachdem, welche Art von Variablen wir visualisieren wollen

Lexical Decision Task (LDT)

- unser erster Datensatz enthält Daten aus einer lexikalischen Entscheidungsaufgabe
- Bei der LDT drücken die Teilnehmer eine Taste, um anzugeben, ob ein Wort ein echtes Wort oder ein Pseudowort ist.



LDT-Variablen

- Die üblichen Variablen, die in einem Experiment zur lexikalischen Entscheidungsaufgabe erhoben werden, sind:
  - Reaktionszeit
  - Genauigkeit (richtig/falsch)
  - Wortkategorie (z. B. real/pseudo, Nomen/Verb)
  - Worthäufigkeit
- Zusätzliche Variablen, die erhoben werden könnten, sind:
  - demografische Daten der Teilnehmer (z. B. Alter, L1/L2, Geschlecht)

`lexdec` Datensatz

- `languageR` ist ein Begleitpaket für das Lehrbuch Baayen (2008)
  - enthält linguistische Datensätze, z.B. `lexdec`.
- der `lexdec`-Datensatz enthält Daten für eine lexikalische Entscheidungsaufgabe im Englischen
  - wir werden mit Variablen wie Reaktionszeiten und Genauigkeit arbeiten

`lexdec`-Variablen

- eine Liste einiger der Variablen ist in Tabelle 1 enthalten

Tabelle 1: Datenwörterbuch für `df_lexdec`: Lexikalische Entscheidungslatenzen, die von 21 Probanden für 79 konkrete englische Substantive erhoben wurden, mit Variablen, die mit dem Subjekt oder dem Wort verknüpft sind.

Variable	Beschreibung
Subject	ein Faktor für die Probanden
RT	ein numerischer Vektor für die Reaktionszeit in Millisekunden
Trial	ein numerischer Vektor für den Rang des Versuchs in der Versuchsliste
Sex	ein Faktor mit den Ausprägungen F (weiblich) und M (männlich)
NativeLanguage	ein Faktor mit den Niveaus English und Other, der zwischen englischen Muttersprachlern und Nicht-Muttersprachlern unterscheidet

LDT-Forschungsfragen

- bevor wir ein Experiment durchführen, haben wir Forschungsfragen, die wir mit den Daten beantworten wollen
  - Wir werden uns heute mit der folgenden Frage beschäftigen:
    - Unterscheiden sich die Reaktionszeiten zwischen Muttersprachlern und Nicht-Muttersprachlern?

Laden der Daten

- unsere Daten sind in dem Paket `lanaugeR` verfügbar, das wir bereits geladen haben
  - um die Daten zu drucken, geben Sie einfach den Namen des Datensatzes ein und führen Sie ihn aus
- Unten sehen wir nur ein paar Variablen, aber Sie sollten mehr in Ihrer Konsole sehen

```
1  lexdec
```

```
  Subject       RT Trial Sex NativeLanguage Correct PrevType PrevCorrect
1      A1 6.340359    23   F        English correct     word     correct
2      A1 6.308098    27   F        English correct  nonword     correct
3      A1 6.349139    29   F        English correct  nonword     correct
4      A1 6.186209    30   F        English correct     word     correct
5      A1 6.025866    32   F        English correct  nonword     correct
6      A1 6.180017    33   F        English correct     word     correct
```

- Wie viele Variablen haben wir? Beobachtungen?

Daten als Objekt speichern

- Um die Daten in unserer Umgebung zu speichern, müssen wir ihnen einen Namen zuweisen
  - Nennen wir es `df_lexdec`, was soviel bedeutet wie “Datenrahmen lexikalische Entscheidung”.

```
1  df_lexdec <- lexdec
```

- jetzt sehen wir es in unserem Enrivonment
  - Doppelklicken Sie darauf, um es im Editorfenster zu sehen.

Relevante Variablen

- Zu den Variablen, die wir haben, gehören:
  1. **Subjekt**: Teilnehmer-ID
  2. **RT**: protokollierte Reaktionszeiten
  3. **NativeLanguage**: die Muttersprache des Teilnehmers
  4. **Word**: welches Wort präsentiert wurde
  5. **Class**: ob das Wort ein Tier oder eine Pflanze war

Aufgabe 1: `?lexdec`

Aufgabe 1

Um herauszufinden, wofür die anderen Variablen stehen, führen Sie `?lexdec` in der Konsole aus.

Erstellen von Plots mit `ggplot2`

- das `tidyverse` ist eine Sammlung von Paketen, die das Aufräumen und die Visualisierung von Daten erleichtern
  - wenn wir `tidyverse` laden, wird diese Sammlung von Paketen automatisch geladen
- das `ggplot2`-Paket ist ein `tidyverse`-Paket, das Plots in Schichten aufbaut

ggplot2 Schichten

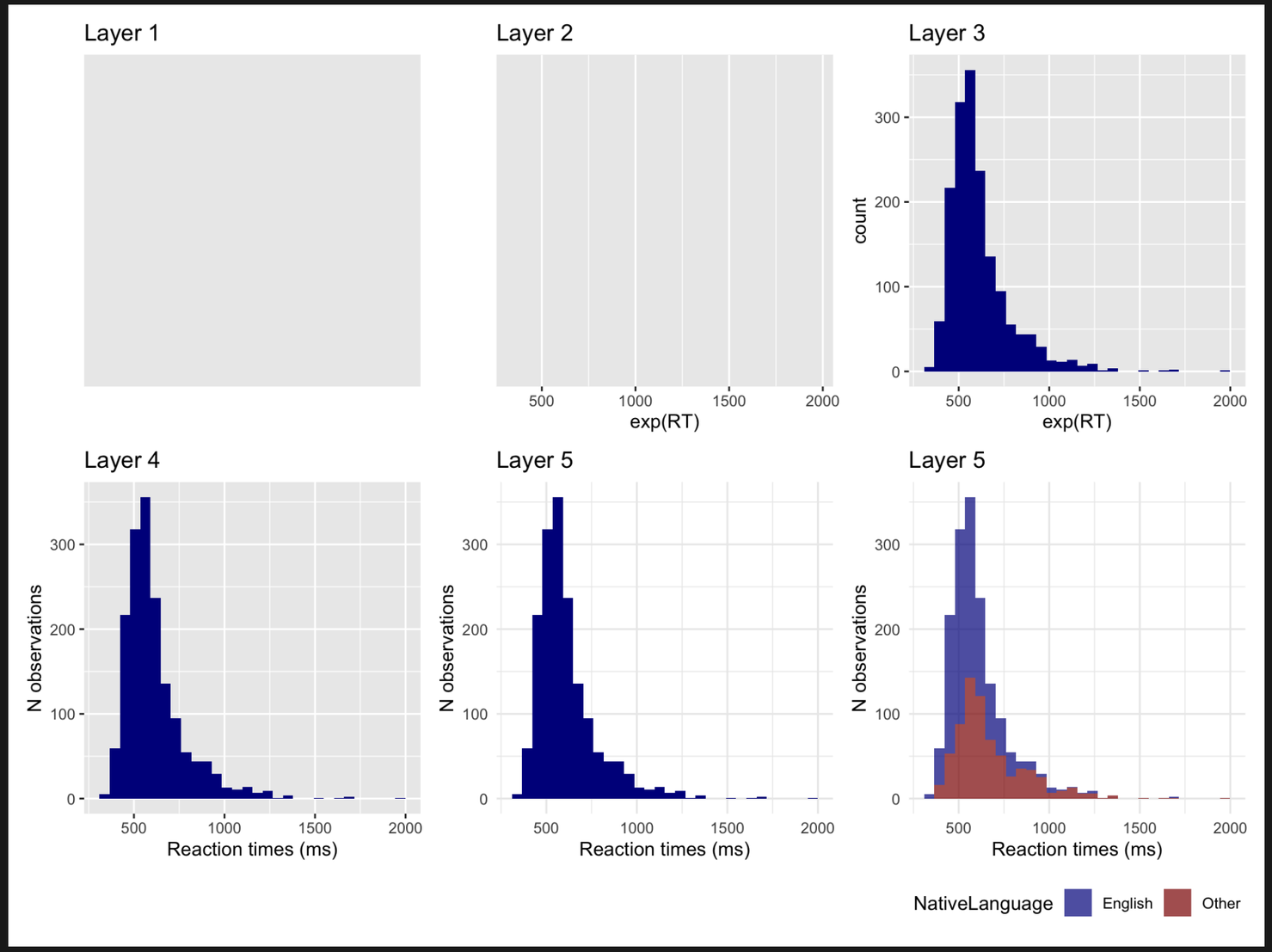


Abbildung 1: Example of layers in a ggplot figure

Ebene 1: leere Leinwand

- die erste Ebene mit der Funktion `ggplot()` ist wie eine leere Leinwand

```
1  ggplot(data = df_lexdec)
```

Ebene 2: Ästhetik der Darstellung

- als nächstes teilen wir `ggplot()` mit, wie unsere Variablen visuell dargestellt werden sollen
  - Wir fügen das “+” am Ende unserer Codezeile ein und verwenden in einer neuen Codezeile die Funktion “aes()”, um unsere *Ästhetik* zu definieren.
- Unsere erste Ästhetik bildet die Reaktionszeiten (RT) auf der x-Achse ab (der untere Teil der Grafik)
  - wir wickeln die protokollierte `RT` in die Funktion `exp()` ein, um RTs in Millisekunden zu erhalten (aus Gründen, die wir nicht diskutieren werden)

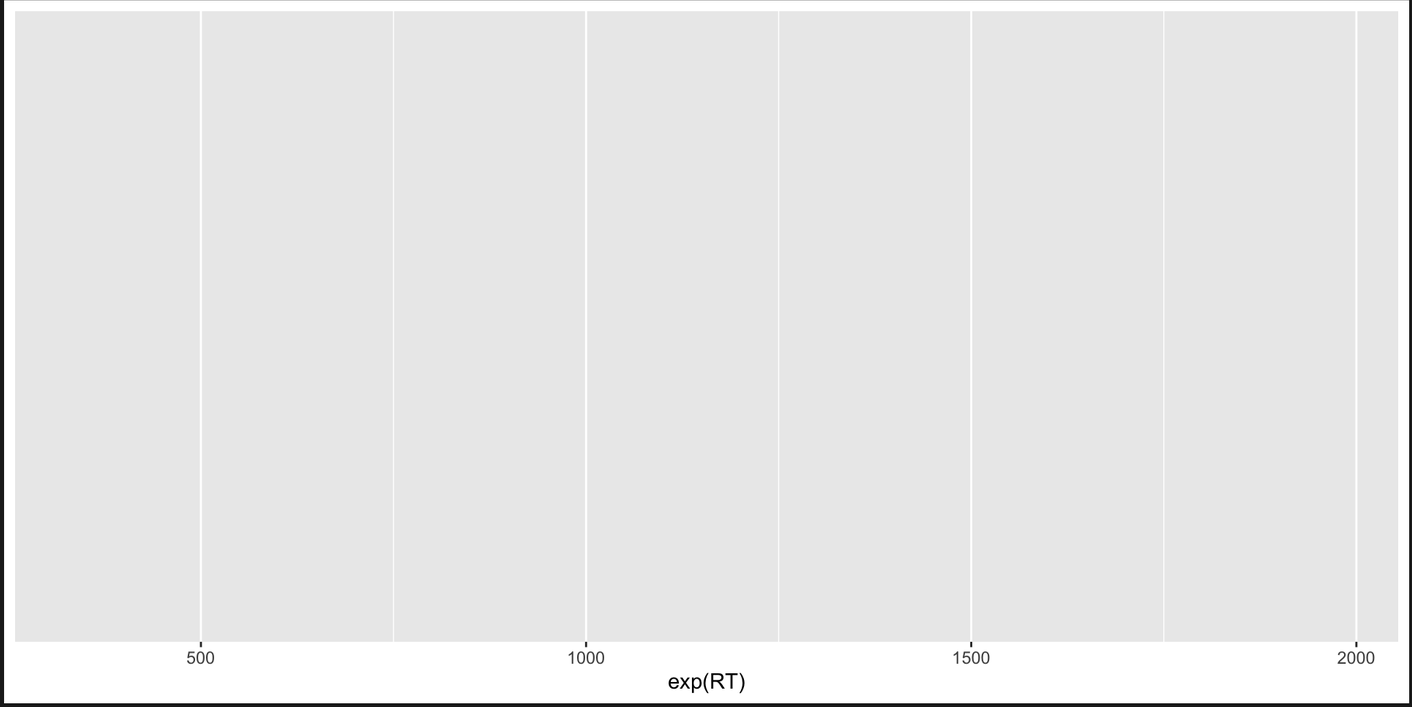
```
1 ggplot(data = df_lexdec) +
2   aes(x = exp(RT))
```

Aufgabe 2: Ästhetische Kartierung

Aufgabe 2

Add the x-axis aesthetic.

Ebene 2: Ästhetik der Darstellung



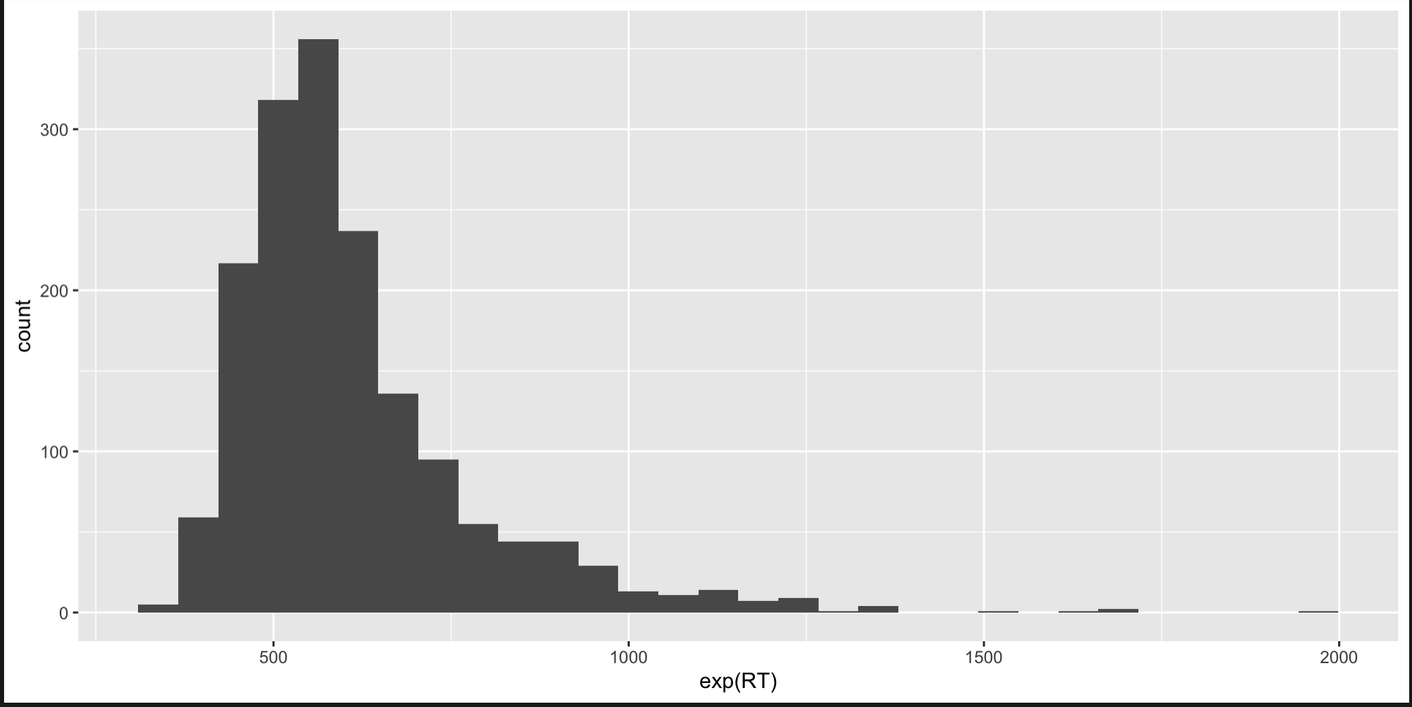
Schicht 3: Hinzufügen von Beobachtungen

- wir sehen keine Beobachtungen (d.h. die Balken) in der Grafik, warum nicht?
  - wir haben `ggplot()` nicht gesagt, wie sie dargestellt werden sollen
- wir müssen ein **Geom** definieren: das *geom*etrische Objekt, das ein Diagramm verwendet, um Daten darzustellen
  - in `ggplot2` beginnen die Geom-Funktionen mit `geom_`
  - wir beschreiben Diagramme oft in Bezug auf die Arten von Geomen, die sie verwenden, z.B. verwenden Balkendiagramme Balkengeome (`geom_bar()`), Liniendiagramme Liniengeome (`geom_line()`), Punktdiagramme ein Punktgeom (`geom_point()`), usw.

Schicht 3: Hinzufügen von Beobachtungen

- Erzeugen wir unser Histogramm mit dem Geom `geom_histogram()`

```
1  ggplot(data = df_lexdec) +
2    aes(x = exp(RT)) +
3    geom_histogram()
```



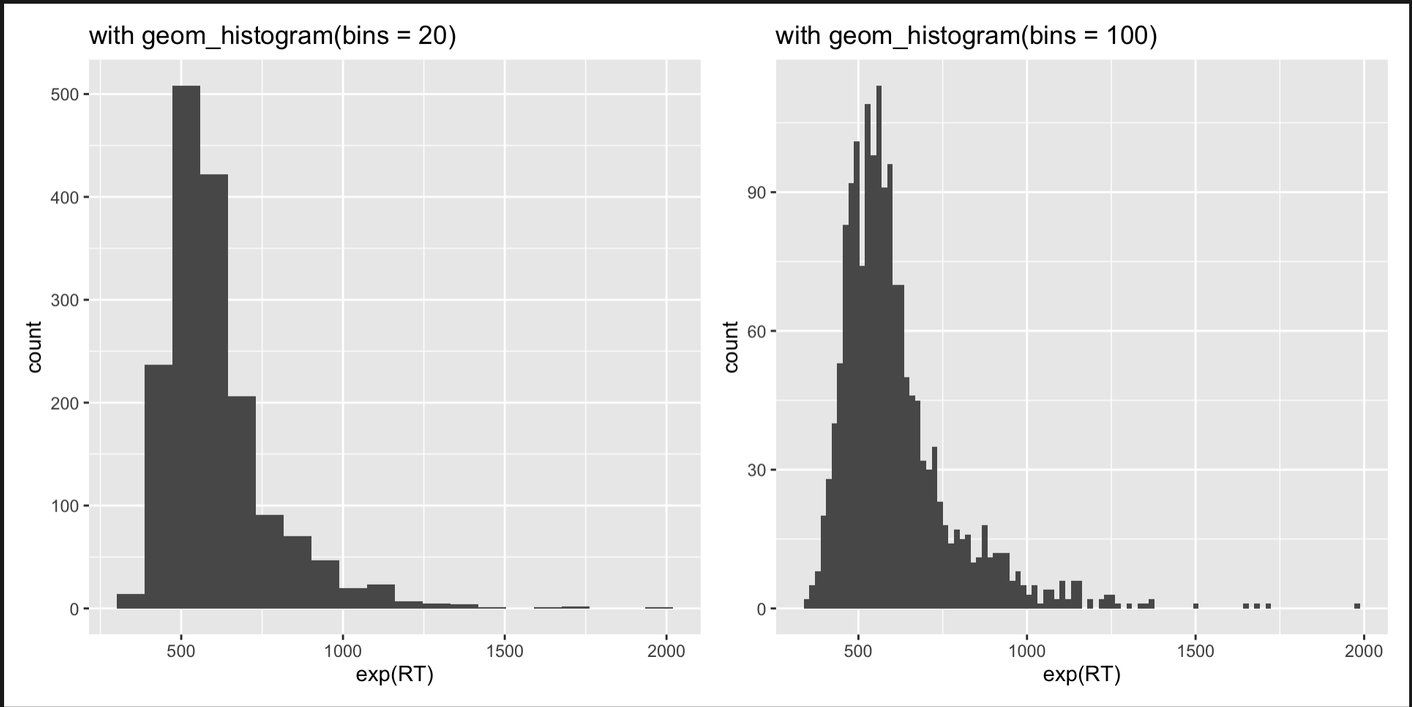
Hinweis

Wir erhielten die folgende Meldung, als wir `geom_point()` einschlossen:

> `stat_bin()` mit `bins = 30`. Wählen Sie einen besseren Wert mit `binwidth`.

Dies sagt uns nur etwas über die Breite unserer Balken: jeder Balken repräsentiert einen Bereich möglicher Reaktionszeitwerte + `bins = 30` bedeutet einfach, dass es 30 Balken gibt, wir können dies ändern und mehr oder weniger Balken haben, indem wir z.B. `bins = 20` oder `bins = 100` in `geom_histogram()` einfügen

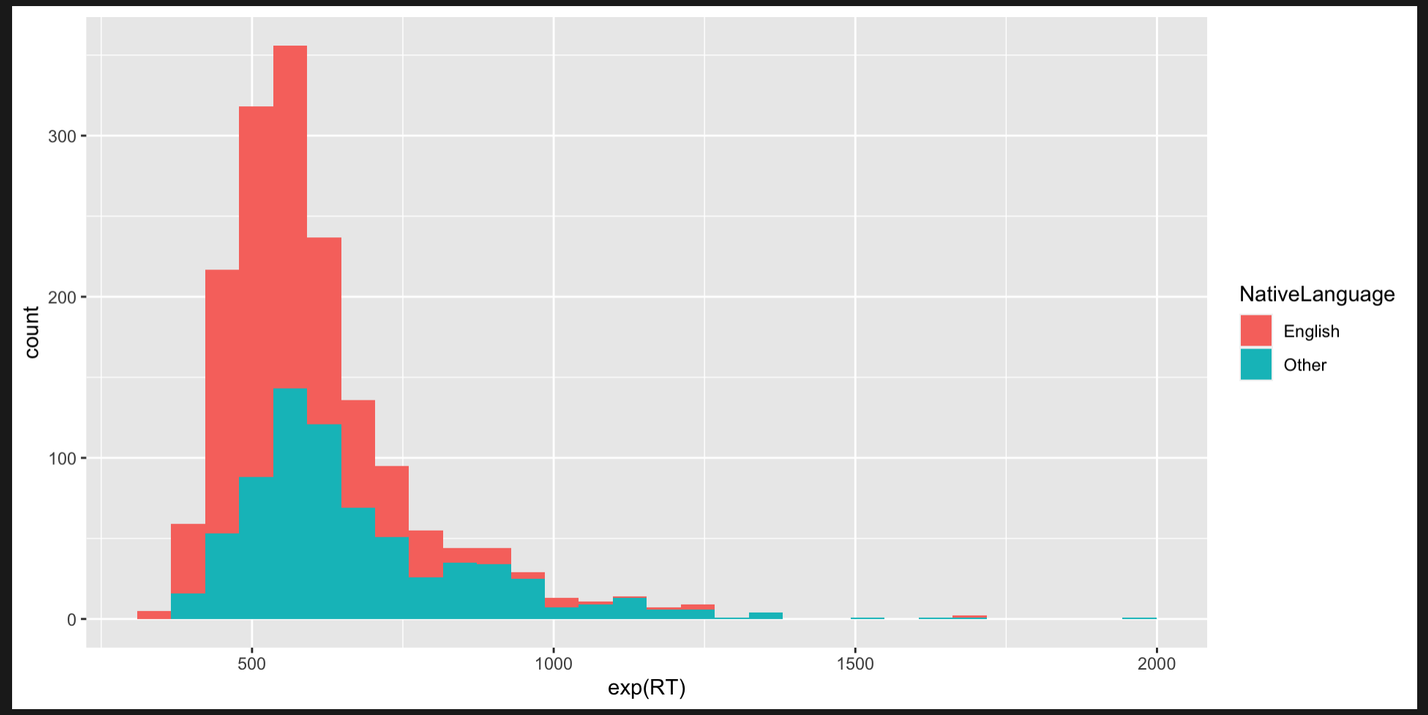
▶ Code



Hinzufügen von Ästhetik

- Es ist nützlich, die Verteilung der Reaktionszeiten im Allgemeinen zu sehen.
  - aber wir wollen normalerweise Gruppen vergleichen
  - z. B. Unterschiede zwischen Muttersprachlern und Nicht-Muttersprachlern oder zwischen verschiedenen Wortarten
- Wir haben auch die Muttersprache als Variable, wie könnten wir diese in unserem Diagramm visualisieren?

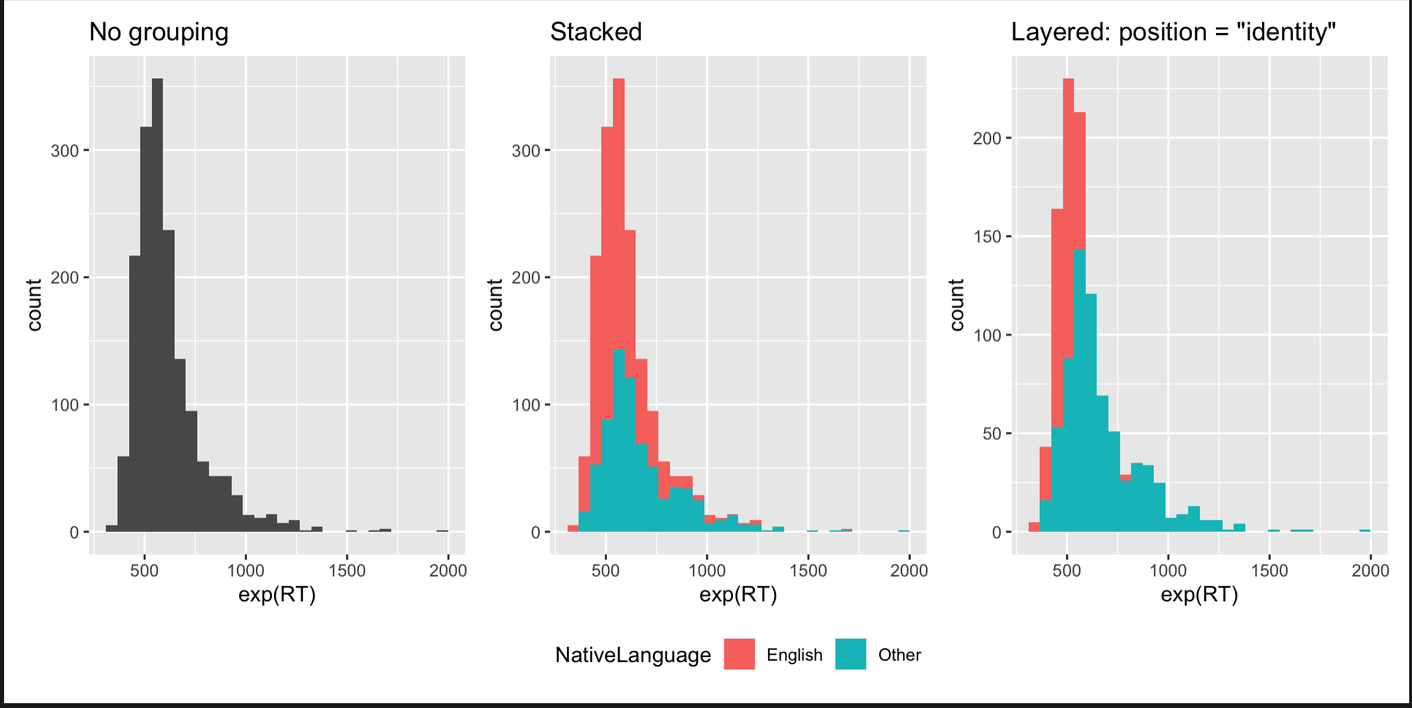
▶ Code



Stapeln versus Schichten von Histogrammen

- wir sehen die roten und die blauen Balken, aber ist das blaue Histogramm über das rote geschichtet?
  - oder sind die roten Balken über den blauen Balken gestapelt?
- Es ist letzteres
  - stellen wir es so ein, dass das blaue Histogramm über dem roten liegt

▶ Code



Globale und lokale Ästhetik

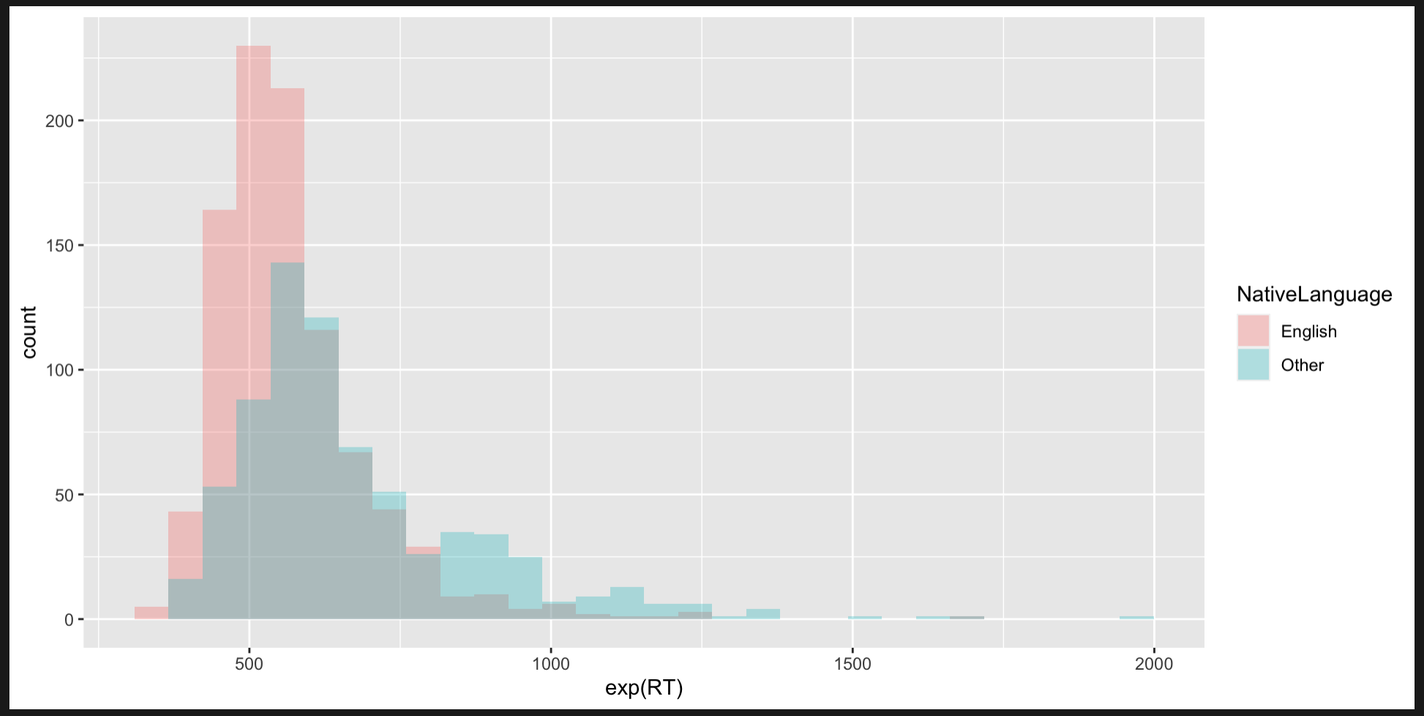
- in unserer endgültigen Darstellung ist die Farbe der Histogramme leicht transparent
  - Wir können dies steuern, indem wir das Argument `alpha = 0.3` zu `geom_histogram()` hinzufügen.
  - alpha kann jeden anderen Wert zwischen 0 und 1 annehmen.

Globale und lokale Ästhetik

Aufgabe 3: Transparenz

Aufgabe 3

Spielen Sie mit der Transparenz des Histogramms geom. Wählen Sie den von Ihnen bevorzugten Alpha-Wert. Die Ausgabe sollte in etwa so aussehen:



Anpassen unseres Plots

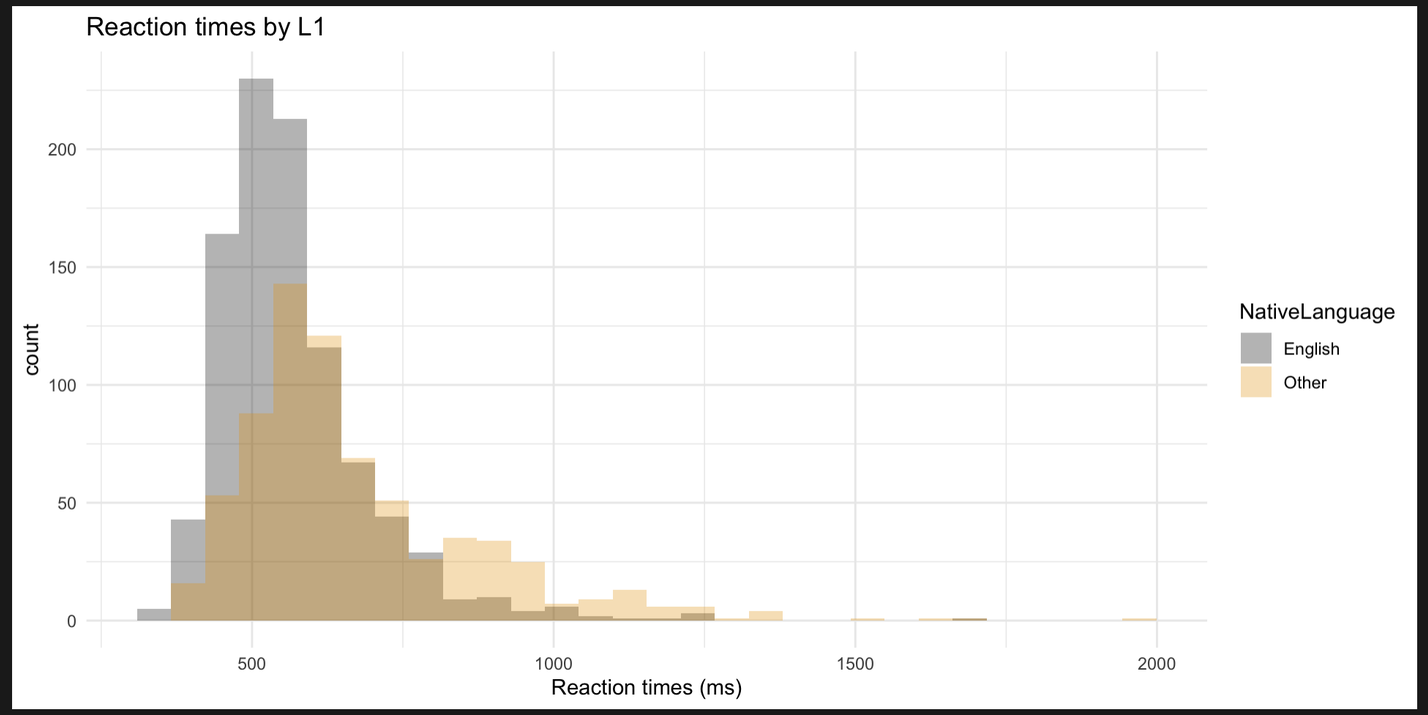
- wir können unsere Achsen- und Legendenbeschriftungen verbessern und auch Titel hinzufügen, indem wir die Funktion `labs()` verwenden
- Wir können auch die Funktion `scale_fill_colorblind()` aus dem Paket `ggthemes` verwenden.
  - dies erzeugt farbenblind-sichere Farben
- Wir werden auch die Funktion `theme_minimal()` aus dem Paket `ggplot2` verwenden; was bewirkt diese Funktion?
- Versuchen Sie, Ihrem Diagramm Folgendes hinzuzufügen
  - Ändern Sie die Beschriftungen entsprechend
  - und fügen Sie dem Code sinnvolle Kommentare mit # hinzu

```
labs(title = "Plot title",
     x = "x-axis label",
     y = "y-axis label") +
  scale_fill_colourblind() +
  theme_minimal()
```

Kommentar

- Der Code und die Darstellung sollten in etwa so aussehen:

```r
# histogram of reaction times by native language
ggplot(data = df_lexdec) +
  aes(x = exp(RT), fill = NativeLanguage) + # set aesthetics
  labs(title = "Reaction times by L1",
       x = "Reaction times (ms)") +
  geom_histogram(position = "identity", alpha = 0.3) +
  scale_fill_colorblind() + # make fill colorblind friendly
  theme_minimal() # set plot theme
```



Speichern von Plots

- Wir können Diagramme in unserer Umgebung speichern, genau wie wir Zahlen und Daten als Objekte speichern können.
  - Sie können Objekte beliebig benennen
  - aber es ist ratsam, den Namen sinnvoll zu gestalten (z.B. *nicht* `fig1` oder `xyz`)
- Nennen wir diese Grafik `fig_lexdec_rt`, für “figure lexical decision task reaction times”.

Aufgabe 4: Figur als Objekt speichern

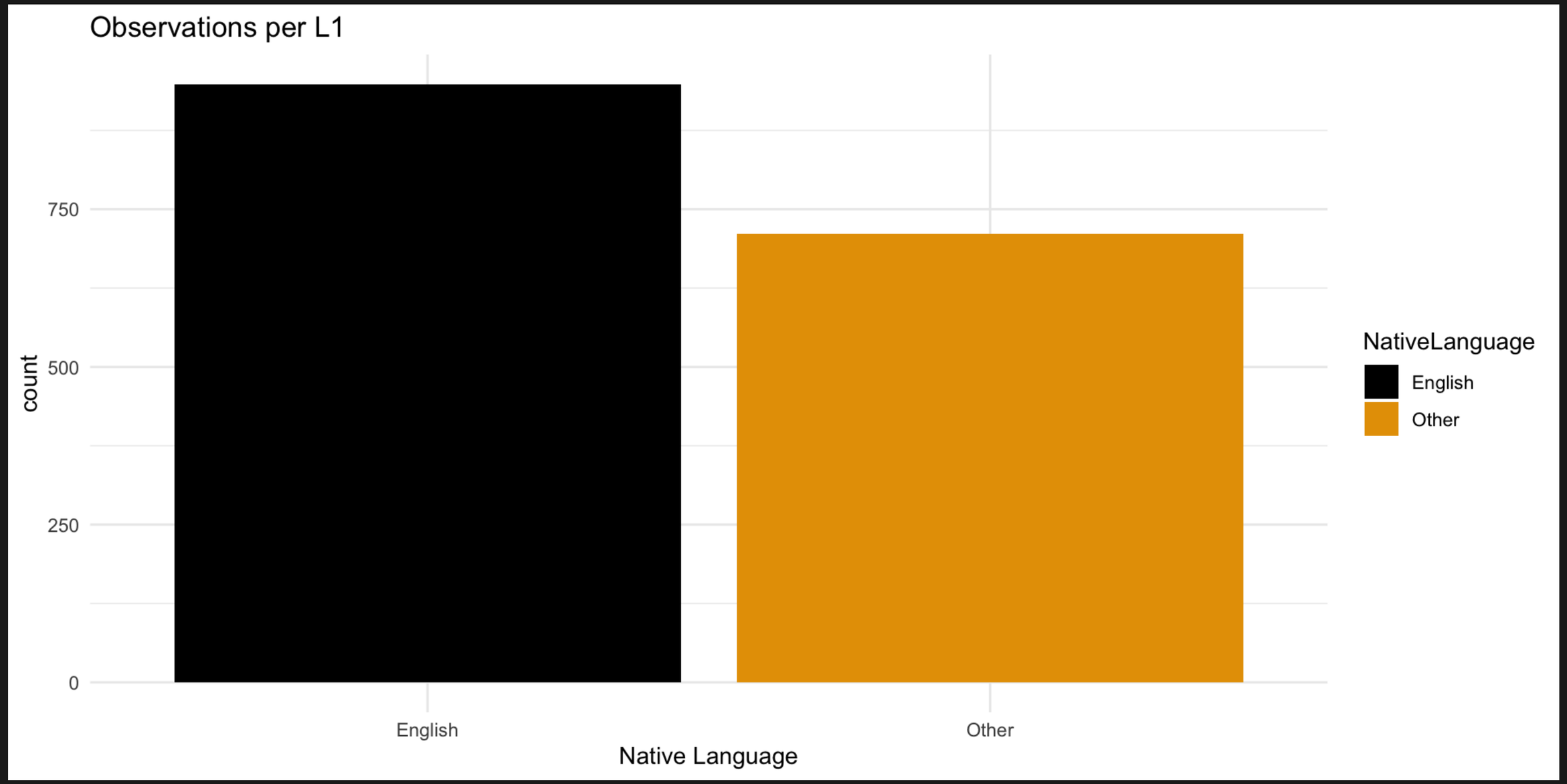
Aufgabe 4

1. Speichern Sie unsere endgültige Darstellung als Objekt mit dem Namen `fig_lexdec_rt`.

Balkendiagramme

1. Kopieren Sie den Code für Ihr Histogramm
2. Nehmen Sie die folgenden Änderungen vor, um unser Balkendiagramm darzustellen
   - Entfernen Sie die Namenszuweisung (`fig_lexdec_rt`)
   - auf der x-Achse wollen wir `NativeLanguage`
   - Ersetzen Sie `geom_histogram()` durch `geom_bar()`
     - Entfernen Sie die Argumente für das Histogramm (kein `position` oder `alpha`)
   - ändern Sie die Beschriftungen entsprechend
3. Speichern Sie das Diagramm als Objekt mit einem aussagekräftigen Namen (z.B. `fig_lexdec_l1`)

- sollte das Diagramm in etwa so aussehen:

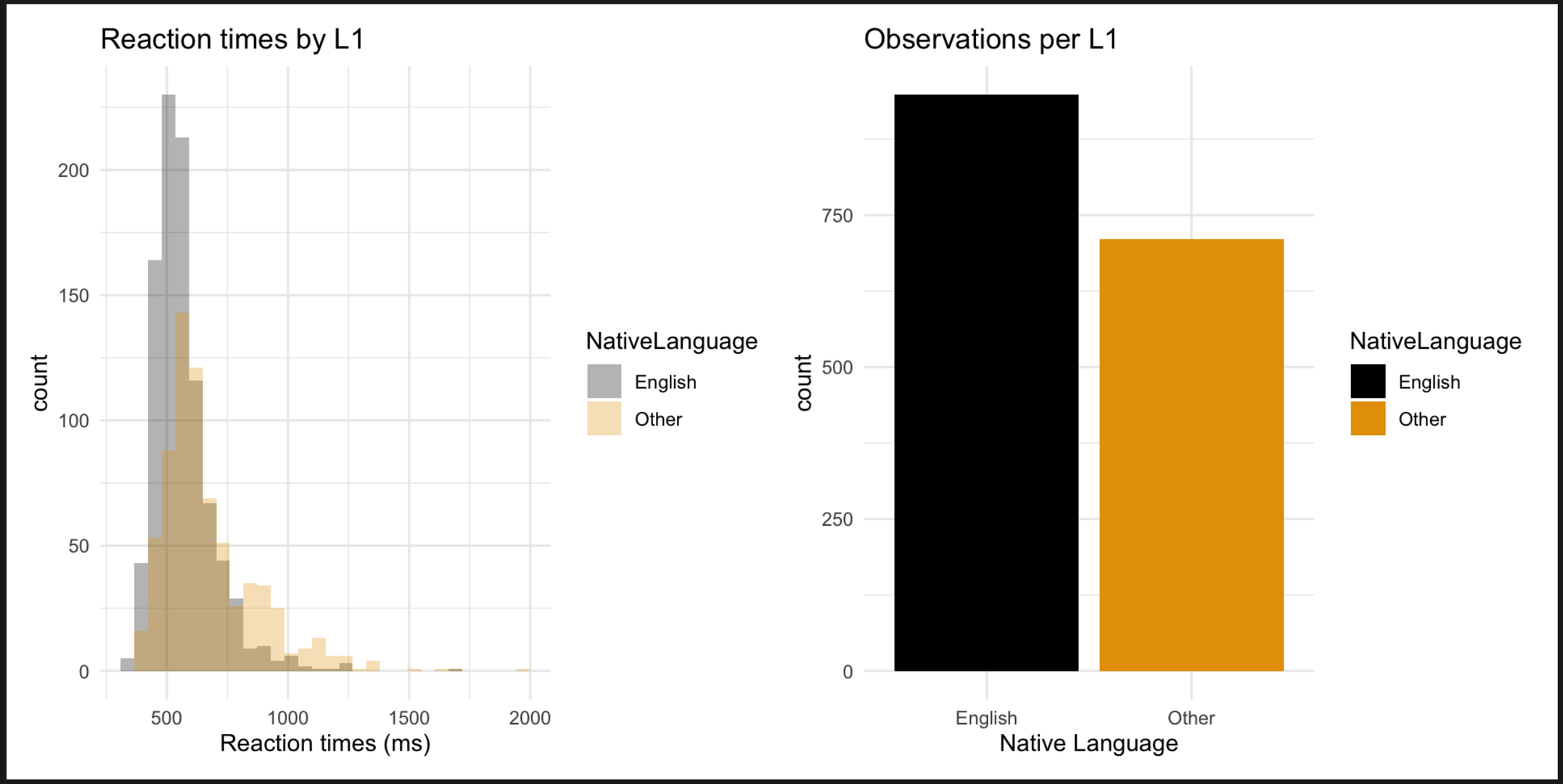


Kombinieren von Plots

- Ein Grund, Ihre Darstellung als Objekt zu speichern, ist, dass wir sie später aufrufen können
  - d.h. Sie können den Plot an einer Stelle in Ihrem Dokument erstellen, sich aber entscheiden, ihn erst im gerenderten Bericht weiter unten zu drucken
- ein weiterer Grund ist, dass wir mehrere Diagramme kombinieren können
  - Dies kann mit einer Vielzahl von Paketen geschehen
  - Versuchen wir es mit dem Paket `patchwork`
    - Benutze `+` um zwei Plots nebeneinander zu verbinden
    - oder `/`, um sie übereinander darzustellen

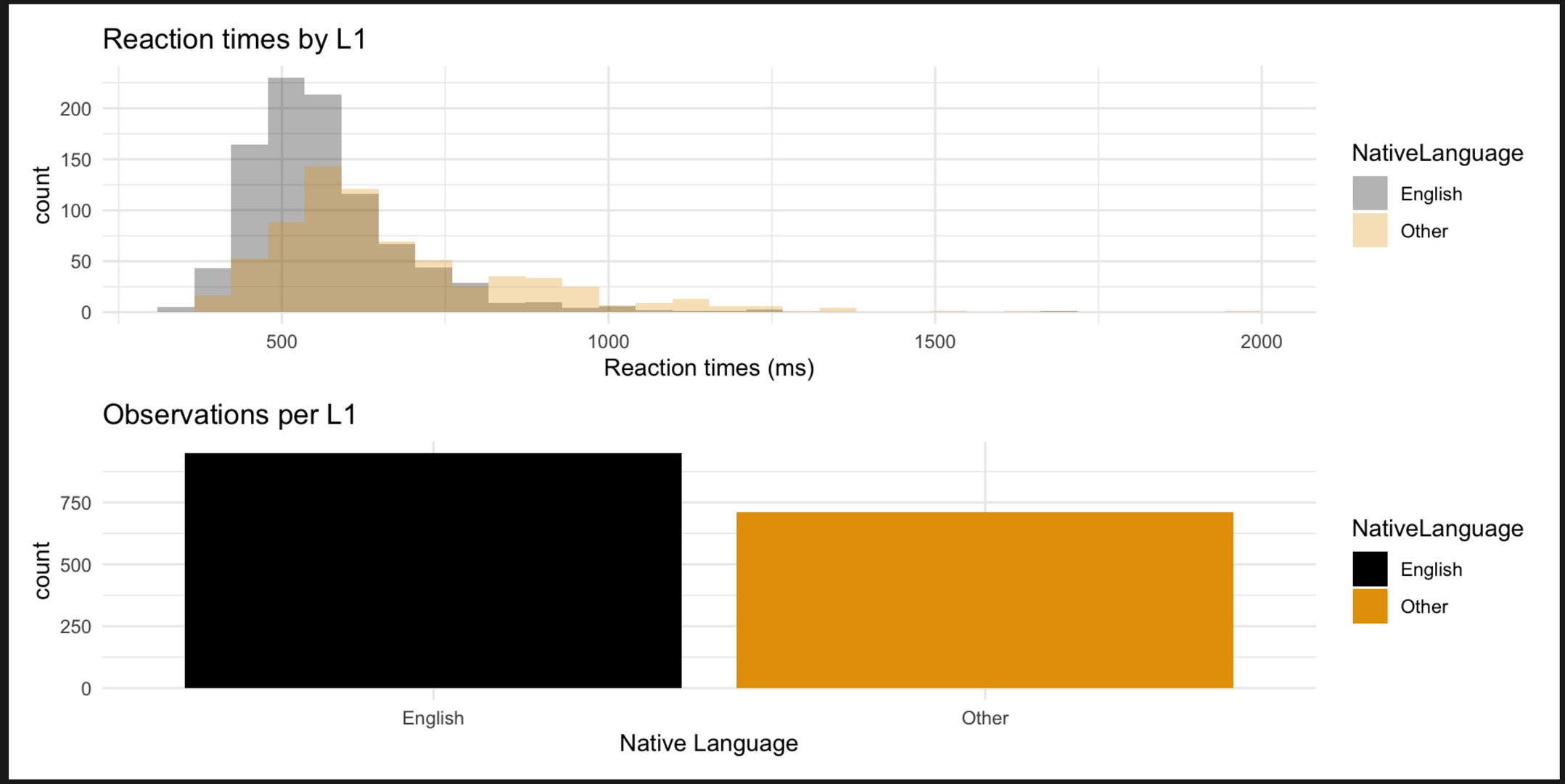
Kombinieren von Plots mit +

```
1  fig_lexdec_rt + fig_lexdec_l1
```



Kombinieren von Plots mit /

```
1  fig_lexdec_rt / fig_lexdec_l1
```



Entscheidung für ein Geom

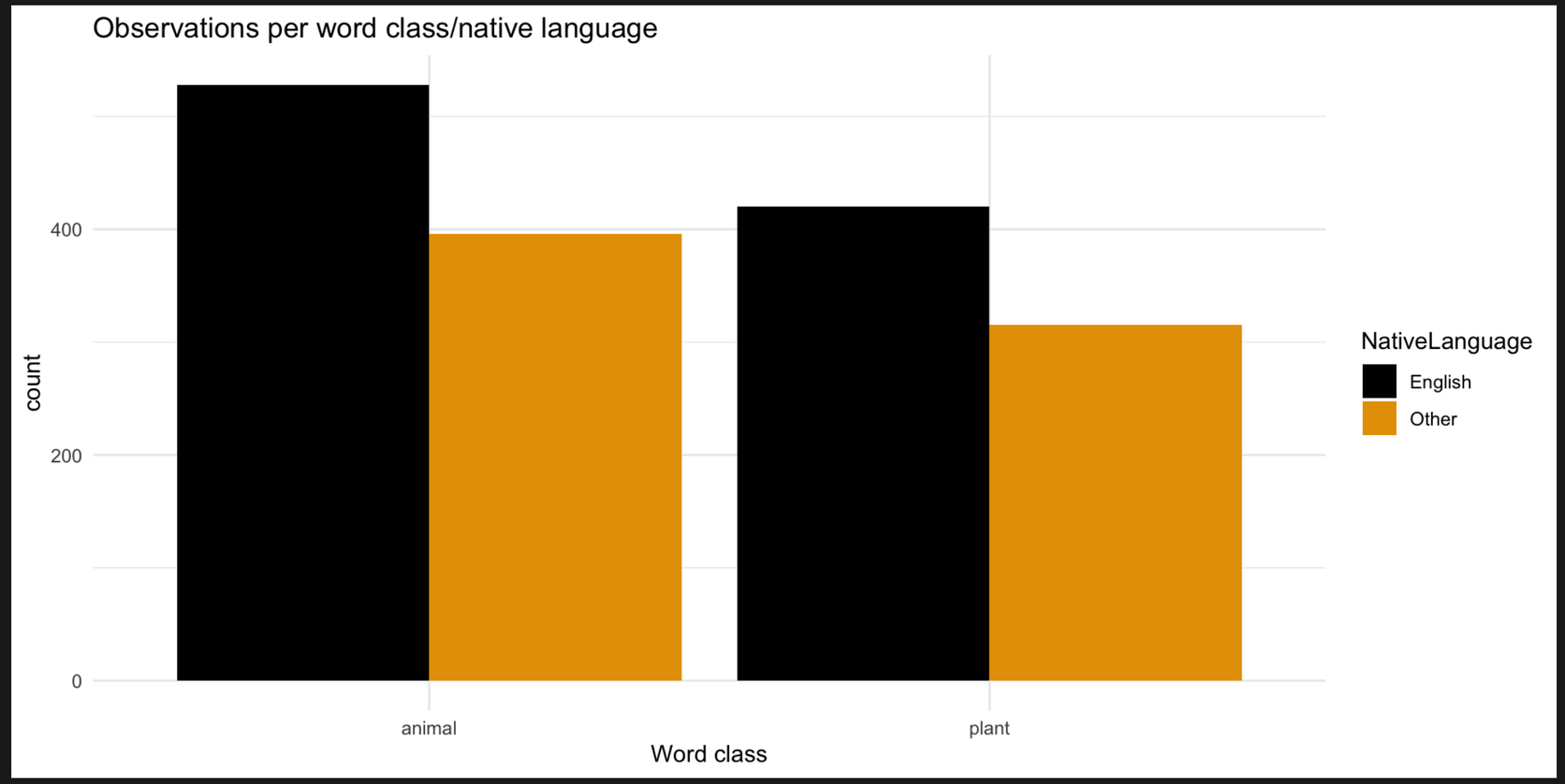
- Warum verwenden wir ein Histogramm für die Reaktionszeit und ein Balkendiagramm für die Muttersprache?
- Um welche Arten von Variablen handelt es sich?
  - Reaktionszeit ist kontinuierlich
  - Muttersprache ist eine kategoriale Variable
- Wir verwenden Histogramme, um die Verteilungen von *kontinuierlichen* Variablen zu visualisieren.
- Wir verwenden Balkendiagramme, um Verteilungen von *kateogrischen* Variablen zu visualisieren.
- Wenn wir wissen, was wir visualisieren wollen (z. B. Verteilungen) und welche Art von Variable wir haben (d. h. kontinuierlich, kategorial), können wir entscheiden, welche Art von Diagramm wir erstellen wollen.
- Oft ist es eine gute Idee, die Darstellung auf Papier zu zeichnen, bevor man in R beginnt (ich mache das auch oft).

Exercises

Diese Übungen sollten auch in Ihrem Skript enthalten sein, wenn Sie es auf Moodle hochladen. Das Durcharbeiten des Unterrichtsmaterials wird Sie auf diese Aufgaben vorbereiten.

1. Reproduzieren Sie unser Histogramm als *Dichte-Diagramm*, indem Sie `geom_histogram()` durch `geom_density()` ersetzen.
   - Was zeigt diese Art der Darstellung?
2. Erstellen Sie ein Balkendiagramm, das die Anzahl der Beobachtungen pro Wortklasse zeigt (Hinweis: Sie benötigen die Variable `Class` aus unserem Datensatz).

3. Drucken Sie Ihren Dichteplot und Ihren Klassen-Balkenplot übereinander mit Hilfe des `patchwork` Pakets
4. Reproduzieren Sie die folgenden Diagramme so genau wie möglich (Hinweis: Sie benötigen das Argument `position = "dodge"`):



Heutige Ziele

Heute haben wir gelernt…

- was Datenrahmen sind
- den Unterschied zwischen kategorialen und kontinuierlichen Daten
- wie man Diagramme mit `ggplot` erstellt
- die richtige Darstellung für unsere Daten auszuwählen

Session Info

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

```
1 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;  LAPACK version 3.11.0

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:
```

Literaturverzeichnis

Baayen, R. H. (2008). *Analyzing Linguistic Data: A Practical Introduction to Statistics Using R*.

Nordmann, E., & DeBruine, L. (2022). *Applied Data Skills*. Zenodo. https://doi.org/10.5281/zenodo.6365078

Wickham, H., Çetinkaya-Rundel, M., & Grolemund, G. (2023). *R for Data Science* (2. Aufl.).