

# Statistics for Linguists

## 08 July 2022

|               |   |
|---------------|---|
| 10:00         | Workshop introduction                     |
| 10:15         | Loading and exploring datasets            |
| 10:45         | Data transformation and coding            |
| 11:15         | Practical exercise                        |
| 12:15         | Review of practical                       |
| 12:30 - 13:30 | LUNCH BREAK                               |
| 13:30         | lmer and glmer                            |
| 14:30         | Post-hoc analysis and model visualization |
| 15:00         | Practical exercise                        |
| 16:00         | Review of practical                       |
| 16:15         | Model building                            |
| 17:00         | End of workshop                           |

# Statistics for Linguists

## Data transformation and coding

# Learning objectives

- You will learn to load/import data
- Explore a dataset and create descriptive statistics
- **Transform a dataset (if needed)**
- **Code your factors**
- Build a mixed model
- Perform post-hoc statistics
- Visualize your data and your model

# 4 main types of data

| Type                | Example                    |
|---------------------|----------------------------|
| numeric             | integer (2), double (2.34) |
| character (strings) | 'tidyverse!'               |
| boolean             | TRUE / FALSE               |
| complex             | 2+0i                       |

## Special types:

|          |                   |
|----------|-------------------|
| NA       | # missing data    |
| NULL     | # empty           |
| -Inf/Inf | # infinite values |
| NaN      | # Not a Number    |

# 4 main types of data

- But what if the automatic coding that R gives isn't correct?
  - For example, participant 1, 2, 3, 4,... often seen as integers, which they are not

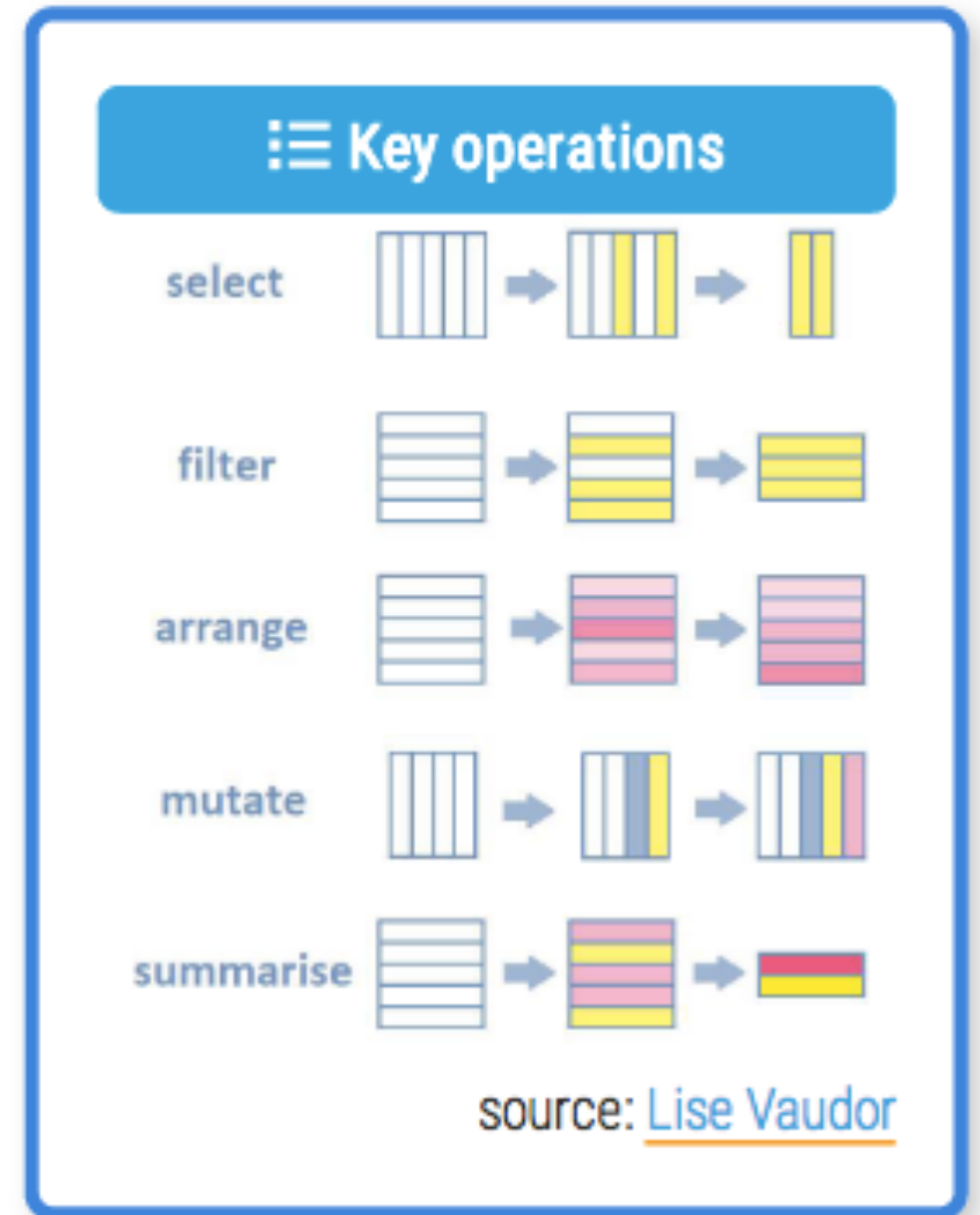
```
> as.character(c(2, TRUE, 'a string'))
```

```
> as.integer()
```

```
> as.factor() #or factor()
```

# Data transformation

Some tidyverse operations



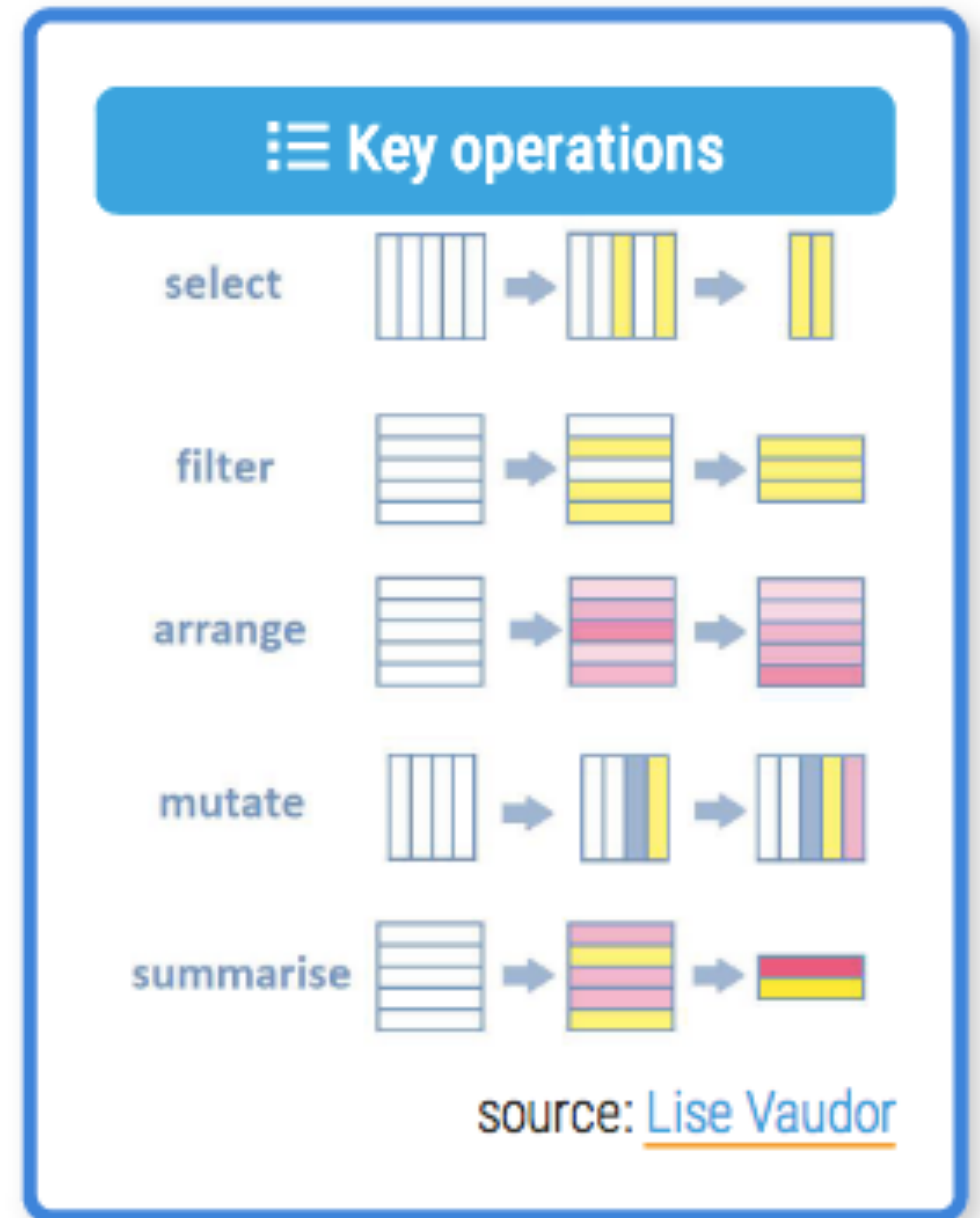
# Data transformation

Some tidyverse operations

```
psycholinguistics_data %>%  
  select(-session, -list)
```

-> type ?select for help

```
psycholinguistics_data %>%  
  filter(ReadingTime < 500)
```



# Data transformation

- Same filtering without tidyverse:

```
> psycholinguistics_data[psycholinguistics_data$ReadingTime < 500,]
```

- This can be useful for, for example, removing outliers



# Data transformation

- Many different ways to select and exclude outliers
  - $2.5 * SD$
  - Based on IQR (per participant?)
  - Based on predetermined values (e.g., an response time of  $< 200$  ms is implausible)
  - we won't go into details of what the best method is here

# Data transformation

- Many different ways to select and exclude outliers
- One simple way is through the boxplot function

```
> boxplot()
```

```
> boxplot()$out
```

- See `?boxplot` for more information

# Data coding

- For categorical variables, factors can be coded in different ways
- Linear models need a baseline: you are in control of setting the baseline for your analyses
- The default coding for factors is treatment or dummy coding: [0,1]

```
> contrasts(psycholinguistics_data$capitalization)
```

```
cap          0
```

```
nocap        1
```

# Data coding

- Treatment coding compares each level of a categorical variable to a reference level. By default, the reference level is the first level of the categorical variable, in alphabetical order.
- You can change the baseline of your model:

```
> psycholinguistics_data$capitalization <-  
  factor(psycholinguistics_data$capitalization,  
        levels=c("nocap","cap"))
```

```
> psycholinguistics_data$capitalization <-  
  factor(psycholinguistics_data$capitalization,  
        levels=c("cap","nocap"))
```

```
> psycholinguistics_data$capitalization =  
  relevel(psycholinguistics_data$capitalization, ref = "nocap")
```

# Data coding

- You can always check your contrasts:

```
> contrasts(psycholinguistics_data$capitalization)
```

```
> levels(psycholinguistics_data$capitalization)
```

- Other contrasts possible. For example, sum or deviation coding compare each level to the grand mean. Compare:

```
> contr.sum(2)
```

```
> contr.treatment(2)
```

# Data coding

- You can always check your contrasts:

```
> contrasts(psycholinguistics_data$capitalization)
```

```
> levels(psycholinguistics_data$capitalization)
```

- Other contrasts possible. For example, sum or deviation coding compare each level to the grand mean. Compare:

```
> contr.sum(2)
```

```
> contr.treatment(2)
```

- This can be useful when you have multiple factor levels or interactions. You can also try

```
> contr.sum(4)
```

# Data coding

- More information:
  - <https://stats.oarc.ucla.edu/r/library/r-library-contrast-coding-systems-for-categorical-variables/>
  - <https://marissabarlaz.github.io/portfolio/contrastcoding/>
- Always check your coding to make sure you're interpreting any model output correctly! This can become complicated when working with factors with multiple levels or interactions

# Data coding

- For non-categorical (i.e. continuous) variables, there are other considerations
- Centering variables is a common way to standardize them, so that the predictors have mean 0. This makes it easier to interpret model outcomes.
- By using `scale(x)` you standardize that variable relative to a normal distribution. This is used when one variable has a scale very different from others



# Data coding

- Mixed models have similar assumptions compared to ANOVAs
- We won't go into all of these in detail, but some normality assumptions may require data manipulations
- **Right (positive) skewed data:**
  - Logarithm  $\log(x)$ . Commonly used transformation
  - Reciprocal  $1/x$ .
- **Left (negative) skewed data:**
  - Square  $x^2$ . Stronger with higher power.
  - Exponential  $e^x$ . Stronger with higher base.

# Data coding

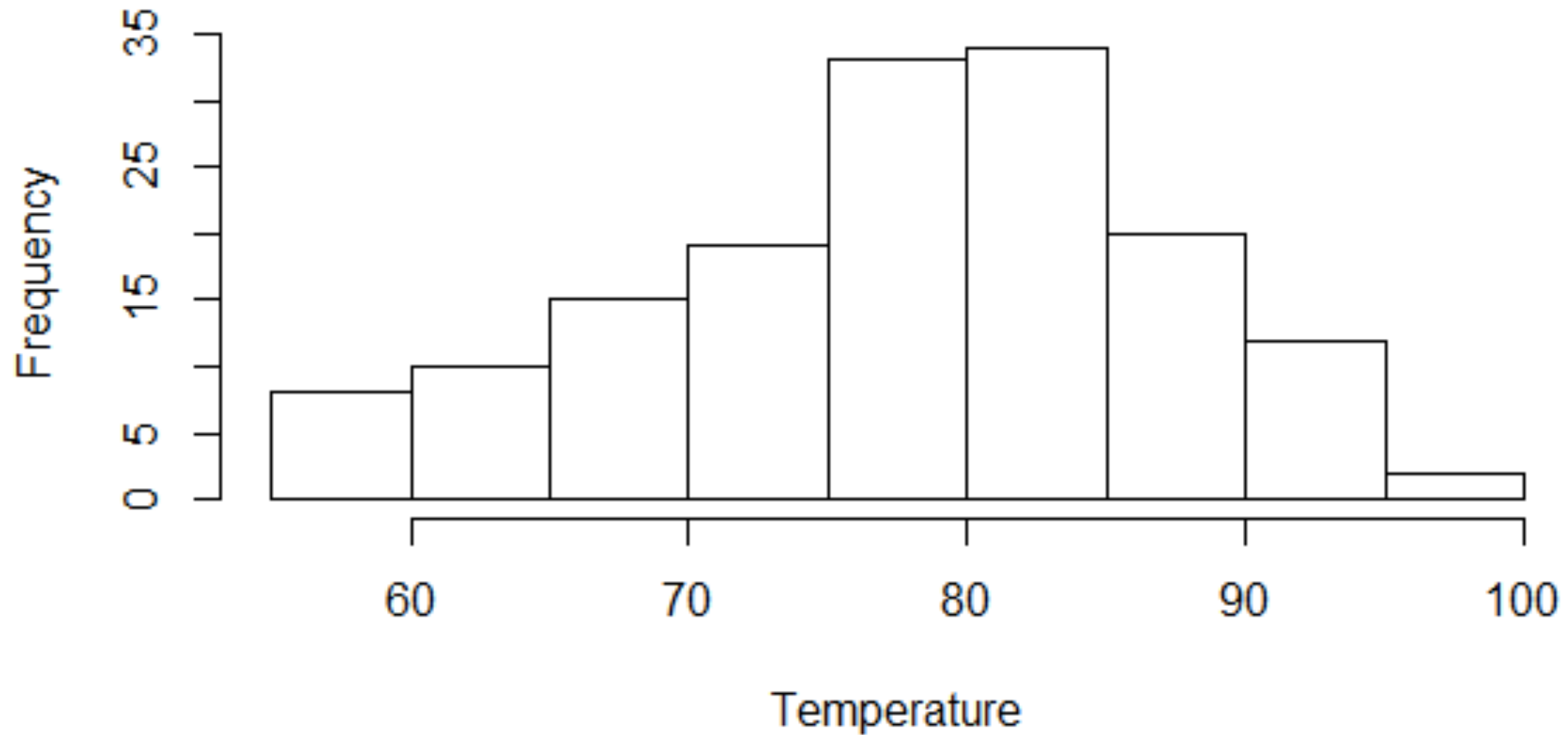
- Mixed models have similar assumptions compared to ANOVAs
- We won't go into all of these in detail, but some normality assumptions may require data manipulations
- To display the distribution, you could use a histogram or a density plot

```
> hist(psycholinguistics_data$ReadingTime)
```

```
> plot(density(psycholinguistics_data$ReadingTime))
```

```
> ggplot(psycholinguistics_data, aes(x=ReadingTime))  
  + geom_density()
```

# Data coding



# Statistics for Linguists

## 08 July 2022

|               |   |
|---------------|---|
| 10:00         | Workshop introduction                     |
| 10:15         | Loading and exploring datasets            |
| 10:45         | Data transformation and coding            |
| 11:15         | Practical exercise                        |
| 12:15         | Review of practical                       |
| 12:30 - 13:30 | LUNCH BREAK                               |
| 13:30         | lmer and glmer                            |
| 14:30         | Post-hoc analysis and model visualization |
| 15:00         | Practical exercise                        |
| 16:00         | Review of practical                       |
| 16:15         | Model building                            |
| 17:00         | End of workshop                           |