# HW 1. Data visualizations

## 0. О том, как выполнять это домашнее задание.

Это задание написано в формате R Markdown (.Rmd). Чтобы облегчить его проверку, мы просим вас писать код в блоки для кода на R, например,

```
# YOUR CODE HERE
# (хештеги используются для комментариев, убирайте хештеги в начале строки, чтобы ваш код запускался)
```

и текст в блоки для текста. Пожалуйста, впишите ниже ваше имя и фамилию:

```
Мое имя:
```

Перед сдачей домашнего задания рекомендуем запустить `Run All` или сгенерировать html- или pdf-страницу с помощью `Knit` , чтобы убедиться, что в финальной версии весь ваш код будет запускаться без проблем.

Файл Rmd ( `HW1.Rmd` ) и сгенерированный из него html (вариант - pdf) вышлите на почту `neurolong@gmail.com` с темой `da4cl` .

# 1. Частотность и фонетика

Во многих лингвистических исследованиях отмечается, что часто используемые в языке слова звучат короче, а при их произнесении наблюдается редукция и коартикуляция. Работа Fabian Tomaschek et al. (https://www.semanticscholar.org/paper/Practice-makes-perfect-%3A-The-consequences-of-for-Tomaschek-Tucker/1e0dbc3787a6da84ffd4c3cae62f1340e4267694) (2018) исследует гипотезу, что моторные навыки произнесения улучшаются с опытом, который, в свою очередь, напрямую связан с частотностью слова. Ученые попросили испытуемых (17 бакалавров университета Тюбингена, 8 мужчин и 9 женщин) прочитать вслух немецкие глаголы, содержащие звук [a:] в основе. Испытуемые были поставлены в экспериментальные условия, которые исподволь заставляли их читать быстрее или медленнее (slow/fast condition).

В этом задании мы просим вас в графическом виде показать распределение в датасете длины звучания всего слова целиком, а также распределение длины звучания интересующего ученых сегмента (звука [a:]) в условиях slow и fast. Хотя логично предположить, что в условии fast произнесение и слов, и сегментов будет короче, все же нужно убедиться, что данные это подтверждают, прежде чем переходить к более сложному анализу по сути вопроса. Кроме того, мы будем уверены, что экспериментальные условия были должным образом соблюдены, ученые не запутались в кодировании данных и документировали результаты корректно.

**Интересующие нас переменные:**

- `LogDurationW` - log-transformed word duration (логарифм длины произнесения слова)
- `LogDurationA` - log-transformed segment duration (логарифм длины произнесения сегмента)
- `Cond` - condition: slow vs. fast (условие).

# 1.1 Загрузка данных

Загрузите пакеты `tidyverse` и `skimr`.
С помощью функции `read_csv` загрузите данные (link (https://raw.githubusercontent.com/LingData2019/LingData2020/master/data/dur_word_frequency.csv)) в переменную `dur_word_freq`.
Используйте функции `summary()`, `glimpse()` и `skim()` (последняя из пакета `skimr`), чтобы изучить структуру данных.

```
#install.packages("skimr")
# YOUR CODE HERE
```

# 1.2 Типы данных

Какие базовые типы переменных (строковые, числовые (непрерывные), целочисленные, логические, комплексные) представляют данные в столбцах `dur_word_freq`?

```
# 1.2
YOUR ANSWER HERE
```

# 1.3 Визуализации ggplot: график плотности (density plot)

О графике плотности распределения можно думать как о сглаженной гистограмме с большим числом столбцов. Общая площадь под графиком составляет 1. Раскомментируйте и запустите следующий код, который позволяет представить распределение непрерывных значений переменной `LogDurationA`, сгруппированных по переменной `Cond` (т. е. длина звука a в условиях fast и slow).

```
# dur_word_freq |>
#  ggplot(aes(x = LogDurationA, group = Cond)) +
#  geom_density()
```

# 1.4

Добавьте в график тему `theme_classic()` и полупрозрачность (alpha = 0.5, указывается в геоме плотности).

```
# YOUR CODE HERE
```

# 1.5

Постройте график плотности распределения длины всего слова (переменная `LogDurationW`) для всех данных (без деления по условиям).

```
# YOUR CODE HERE
```

# 1.6 Боксплот (базовый R)

Раскомментируйте и запустите следующий код.

```
# boxplot(LogDurationA ~ Cond, data=dur_word_freq)
```

Результатом будут так называемые “ящики с усами” (*box and whisker plot*) для значений переменной `LogDurationA`, сгруппированным по переменной `Cond`. Больше об этом типе визуализаций можно прочесть в Википедии (https://ru.wikipedia.org/wiki/Ящик_с_усами) (или Английской Википедии (https://en.wikipedia.org/wiki/Box_plot).

## 1.7

Сравните положение медианных значений `LogDurationA`, а также значений 1-го и 3-го квартиля в условиях fast и slow, запишите кратко ваши выводы ниже (2-3 предложения).

```
# 1.7
YOUR ANSWER HERE
```

## 1.8. Боксплот в ggplot2

Для переменной `LogDurationW` представим боксплоты в условиях fast и slow с помощью пакета `ggplot2` (геом `geom_boxplot`). По оси X у вас будут заданы два боксплота (задаются переменной `Cond`). По оси Y будет представлено распределение длин в переменной `LogDurationW`. Заливка должна зависеть от условия (переменной `Cond`). Все эти три аргумента задаются в базовой эстетике ggplot.
Измените (сделайте более понятными) подписи осей X и Y, добавьте тему `theme_classic()`.

```
# YOUR CODE HERE
```

Необязательное задание - измените цвет заполнения боксплотов с помощью `scale_fill_brewer` из пакета `RColorBrewer` (палитра Dark2).

## 1.9 Скрипичный плот в ggplot2

С помощью ggplot2 постройте скрипичные графики (violin plot, геом `geom_violin`) для переменной `LogDurationA` в двух условиях fast и slow.

```
# YOUR CODE HERE
```

## 1.10 Jitter

К предыдущему графику 1.9 новым слоем добавьте конкретные точки из вашего датасета, при этом используйте jitter (геом `geom_jitter`), чтобы немного развести точки в стороны.

```
# YOUR CODE HERE
```

## 1.11. Скаттерплот в ggplot2

Постройте диаграмму рассеяния (= точечную диаграмму = скаттерплот, геом `geom_point`) для переменных `LogDurationA` (по оси X) и `LogDurationW` (по оси Y). Используйте прозрачность 0.3. Добавьте на график регрессионную прямую (метод “lm”).

```
# YOUR CODE HERE
```